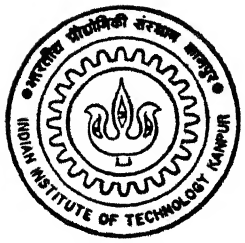
ORDER PICKUP AND STORAGE SYSTEM WITH DUAL COMMAND CYCLE IN A WAREHOUSE

by

Krishna Chandra Jha



DEPARTMENT OF INDUSTRIAL AND MANAGEMENT ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

March, 1996

ORDER PICK UP AND STORAGE SYSTEM WITH DUAL COMMAND CYCLE IN A WAREHOUSE

A Thesis Submitted

in Partial Fulfillment of the Requirements

for the Degree of

Master of Technology

by

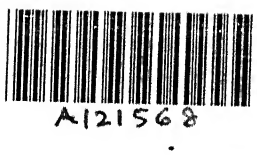
Krishna Chandra Jha

to the

DEPARTMENT OF INDUSTRIAL & MANAGEMENT ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

MARCH, 1996



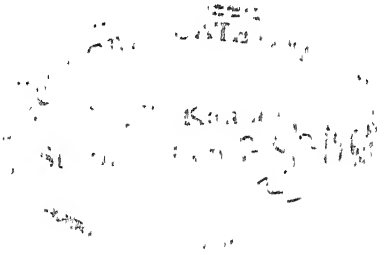
IME-1996-M-JHA-CRD.

CERTIFICATE

This is to certify that the present work on " Order pick up and storage system with dual command cycle in a warehouse ", by Krishna Chandra Jha has been carried out under my supervision and has not been submitted elsewhere for award of a degree

(A. K. Mittal)
Professor
Industrial & Management Engineering
Indian Institute of Technology
Kanpur- 208 016

March, 1996



ACKNOWLEDGEMENTS

First and foremost I owe my heartfelt thanks to Dr. A K Mittal despite his demanding schedule as Dean Planning and Resource Generation, he made himself available whenever I required his guidance.

I am also thankful to my colleagues Kansal, Puneet, Major Ghosh, Deepak, Kapil, Shobhit, Parvesh, Rishi, Madhav, and Gopal for their invaluable support in carrying out this work. I am also grateful to all members of IME family for their valuable suggestions, direct or indirect, during the course of my thesis.

March, 1996 Krishna Chandra Jha

ABSTRACT

In this dissertation an attempt has been made to solve the problem of storage and retrieval in a warehouse using dual command cycle

Different cases have been identified on the basis of factors such as single item storage/retrieval or multiple item storage/retrieval system, type of the equipment as single bin carrying equipment or a multiple bin carrying equipment, etc. The structure of these problem is examined and the exact methods as well as heuristics are developed for each of the cases

The performance analysis of different heuristics with exact methods are done for different factors which may affect the performance of the methodologies. The major factors considered for performance analysis are the number of storage locations, number of retrieval locations, and the capacity of the S/R machine.

In the case when the equipment can carry only one item, the heuristics proposed turned out to be very efficient as well as provided solutions very close to the optimal solution for a large percentage of the problem. However in case of trolley problem, the heuristic developed did not perform well for all the cases.

CONTENTS

List of Figures

CHAPTER 1

INTRODUCTION

1.1 Automated Warehousing System

Historically warehousing was considered to be a work of arts, but the development in the manufacturing units forced it to be considered scientifically and methodologically. In the last twenty years, researchers have begun to develop principles and procedures which enable management to solve some of the problems in logistics(i e, warehouse location).[editorial EJOR - 1992].

Today's environment is characterised by customer markets, JIT supply relations and high service and quality requirements Customers want short lead times, multi frequency and variable deliveries, their needs are ever more sophisticated. This demands flexibility, efficiency and total quality of the warehousing operations. In this environment warehousing and order picking play a key role in the distribution and manufacturing processes.

An article in the Journal of Industrial Engineering claims that the sales of Automated Storage/Retrieval Systems(AS/RS) are expected to rise by 500%. This expected increase, coupled with a study by Enyan and Rosenblatt[1994] showing that material handling costs may run up to about 30% of the production costs, makes the subject of AS/RS increasingly important.

All these factors and computational developments leads the all-round development in warehousing which assists a manager in reducing the material handling costs in the warehouse, in determining the optimal stock assignment and order picking schedule.

The functions performed by the warehouse are[Agarwal K et al., 1995]

[1] Receiving the goods from the stores

[2] Storing the goods until they are required

[3] Picking the goods when they are required

[4] Shipping the goods to appropriate users

The plans to satisfy above functions must be dynamic to cope with the technological developments, resource crunch and market forces. A successful warehouse maximizes the effective use of the warehouse resources while satisfying customer requirements, which enforces the management to meet the following objectives.

[1] Maximize the effective use of space

[2] Maximize the effective use of equipment

[3] Maximize the effective use of labor

[4] Maximize the accessibility of all items

[5] Maximize protection of all items

The work on this dissertation is motivated by the above developments and the importance of warehouse design for a practical manufacturing unit. This work is in continuation of the works done by Reddy N. J. M.[1994], Agarwal K.[1995], Bansal R [1995] and Jain R. K.[1995], whose works can be summarised as follows:

I. Reddy[1994] has dealt with the selection and optimal utilization of material handling equipment. In this, Reddy has developed algorithms to link the production schedule(Batch Sequences) into the requirements of part inventory through the concept of due date(last delivery time) at the work station. The end output of these algorithms is to provide an order pick up schedule(equipment wise) with latest time of filling the order at the output dock of the warehouse.

II Agarwal K [1995] has extended Reddy's work and considered a different module of the storage and retrieval costs, incurred while storing and retrieving various items inside the central warehouse, are to be minimized. in this dissertation an on-line information system has been developed to facilitate efficient and fast operation of the warehouse

III. Bansal R.[1995] has worked on the practical problems naming Stock Assignment Problem and Order Pick up Problem in a warehouse. He has developed an on-line information system which takes care of the warehouse status at every point of time. He has developed some heuristics for both of the cases

IV. Jain R K [1995] has extended Agarwal's thesis and has considered the problem of simultaneous location and routing related to storage assignment in a warehouse. He has considered rectangular warehouse, which may be of single aisle or multi-aisle type. He worked on single command case and has solved the problem in polynomial time by dynamic programming formulation.

The work in this dissertation emphasises on the dual command case, in which the items can be stored and retrieved in the single tour of the equipment. The problem resembles to class of problem better known as location routing problems[Golden B. L and Assad A. A.. 1988].

1.2 Literature Review

The stock assignment in a warehouse is basically divided into two parts. One which we call dedicated storage policy and other is shared storage policy. Now once the allocation of items between the reserve and picking areas has been decided upon, the items must then be assigned to storage locations. In a dedicated storage policy, a set of storage locations is reserved for each product for the duration of the planning horizon. Furthermore, since the same priority is given to all units of a product, these units are assigned to consecutive location in the warehouse.

A shared storage policy, on the other hand, allows units of different products to successively occupy the same location While the literature deals mostly with the dedicated storage policies, results concerning shared storage systems have recently been published Goestchalckx and Ratliff[1990] demonstrate that shared storage policies specifically those which assign storage locations on the basis of the duration-of-stay of individual units, yields significant reduction in the rack size and travel time relative to dedicated storage systems [Cormier et al , 1992]

The utilization and efficiency of a warehouse is much dependent on the structure of aisle Tretheway et al [1994] have discussed different modules and have developed algorithms for the finding the location of aisles and their structure Their algorithms integrates aisle location into a solution methodology which has minimization of material, flow cost as its objective In their procedure, an aisle structure is fixed and then the departments are located around the aisle structure Assumed input to the procedure is a cost-of-flow per unit distance matrix, the required areas of the departments, the dimensions of the plant, and the aisle structure desired As discussed by J Ashayeri and L F Gelders[1985], the warehouse design problem can be divided into two parts, one deals with the warehouse layout and/or selection of the type of the warehouse, the other deals with the operational policies of a warehouse system The later includes the research in determining the warehouse configuration and layout which minimizes material handling and building perimeter costs

The history of storage problem goes back to Kind[1965], who demonstrated that the space utilization should be considered not only at the peak inventory, but over the entire inventory cycle of the product Kind assumed a randomized storage system, FIFO lot rotation and unit load storage and retrievals Ten years later he revised his earlier work

and concluded that using pallet racks is more attractive for increasing space utilization and decreasing investment costs

1.2.1 Location - Routing Problem

Location-Routing Problems(LRPs) involve locating a number of facilities among candidate sites and establishing delivery routes to a set of users i such a way that total system cost is minimized Essentially LRPs are Vehicle Routing Problems(VRPs) in which optimal depot locations and route designs must be decided

Technically speaking location - routing problem can be stated as

Given a feasible set of potential depot sites and customer sites, find the location of the depots and the routes to customers from the depots such that the overall "cost" is minimized The overall cost is the sum of the location and distribution cost [Shrivastva et al , 1990]

Laporte G and Nobert Y [1981] in one paper showed that de linking locational decisions with routing decisions may result in sub optimal solution Nevertheless, LRP is a large and complex problem, which cannot be solved directly using existing mixed integer programming techniques

Nambial et al [1981] proposed two heuristics for solving a large scale specific problem of locating central rubber processing factories to process rubber collected daily from a number of collecting stations

Jain R K [1995] has shown in his thesis that using the framework presented in 3-layer LRP, warehousing problem can be modeled as two-layered version of it He has also shown that although the LRP and warehousing problem have completely different environment, but because of the very purpose(i e cost optimization, locations and tours optimization) being same, similar procedures for solving both the problem can be adopted

1.2.2 Optimal Allocation of Resources

The throughput capacity of an automated warehouse primarily depends on two factors, the number of available storage spaces and the efficiency of the storage and retrieval system Taking this in account, Azadivar[1987] has developed a stochastic optimization method to allocate resources between the random access and rack storage spaces Random access spaces are assumed to require more resources per unit to provide, but they contribute more to the efficiency of the operation of the system because items can be stored in and retrieved from them without delay Rack spaces, on the other hand, need the services of a stacking crane for storage and retrieval As a result, the queuing system formed limits their utilization Azavidar has presented a method for the optimum allocation of the resources among these two types of spaces so that the overall throughput capacity of the warehouse is maximized

There may be the case when the capacity of the AS/RS is insufficient to store all items The warehouse management must then decide which items to assign to AS/RS and in what quantities they should be stored Hackman and Rosenblatt[1990] have developed a heuristic, which decides which items to be stored in so called primary location and which in secondary location In this method, all items are assigned a primary location, from where customer requests are fulfilled Items assigned a primary location within the AS/RS are also assigned a secondary location in central storage The process of transferring stock from the secondary location within central storage to the primary location within the AS/RS is called an internal replenishment The relevant consideration in their heuristics are the savings in time and labor whenever an item is retrieved from the AS/RS(as compared to the original manual procedure), and the cost to replenish the AS/RS from central storage

1.2.3 Dwell Point Rules

In the operation of an automated storage/retrieval system, several policies to strategically position the storage/retrieval machine when idle to reduce travel time in warehouse operation have been advocated Egbelu[1991] developed two separate models based on linear programming formulations which are suitable for dynamic control of an AS/RS, among these, one model is based on the minimization of the expected travel time while the other is based on the minimization of the maximum travel time between points Hwang & Lim[1993] have proposed an algorithm for the Egbelu models, utilising well known results in location theory They showed that one of Egbelu's models can be transferred into a single-facility location problem trough two consecutive linear transformations whose solution is well known Also by utilising a linear transformation and a simple minimax property, the other model is reformulated into such a form that its solution is easily found Besides these two theories, there are also theories mainly proposed by Bozer and White[1984], Goetshalcks[1983], Graves et al [1977], Hansman et al [1976], Tompkins and White[1984] etc Egbelu and Wu[1993] have done simulation to compare six theories Two of them are proposed by Egbelu himself Others are referred as LP expect and LP minimax

These rules use linear programming models to determine the dwell point The analysis indicates that LP minimax are effective strategies to specify dwell point But it is also shown that as the traffic rate increases, dominance is switched to a strategy referred to as Input Strategy In Input strategy, the S/R machine is always positioned at the input point whenever idle In this dissertation this strategy has been adopted for all purposes

1.2.4 Class - based Rectangular AS/RS

Enyan and Rosenblatt[1994] have discussed class-based rectangular AS/RS A rectangular in time warehouse is the warehouse in which for assumed rack, the time to reach most distant row will be different than to reach most distant column They have considered the single command cycle and when the demand, or turnover frequency, of each item is constant and known

In the class-based policy the items are divide into groups according to their demand curves At the same time, the rectangular warehouse is divided into regions, where the numbers of regions is equal to the number of groups Then the allocation of groups of items to classes[regions] in the warehouse[rack] is established The regions closest to the I/O point is assigned the group of items with the highest turnover rate, and so on They have divided the regions in three types, namely, square regions, rectangular regions and transient regions They have developed the algorithms for two class and three class warehouse and then have generalized it for other cases

In this dissertation, this approach has been considered in terms of grouping the items for aisles Hence the inter-aisle movement can be prevented by this approach Another application of this method is in the relocation of items within warehouse Jaikumar and Soloman[1990] have developed the method for optimal relocation of pallets with a high expectancy of retrieval within each storage rack of an automated warehouse to meet the fluctuating, short term throughput requirements imposed on the automated storage-retrieval machines

The prepositioning of these pallets closer to the input/output point of each rack during off-peak periods will reduce the expected travel time for the storage/retrieval machines

during future peak periods of the planning horizon Their operational objectives are a) the long term assignment of pallets to storage locations to meet throughput requirements and b) the determination of the number of relocations to be made in the warehouse and the pallets to be relocated during each planning period They again considered class based policy and made the relocation by moving items distant classes to closer classes

In contrast to above case, Jarvis and Mcdowell[1991] have taken the case of order picking warehouse, and have shown that if the aisles are not symmetrically located about the dock, then simply assigning the most frequently picked items to the nearest aisles will not necessarily minimize average travel distance They have presented the necessary conditions for optimally locating products They have also developed a heuristic based upon these conditions

1.2.5 Dual Command Cycle and Sequencing Retrievals

The work in this dissertation is on sequencing of the storing and retrieving items, hence here we will like to discuss the research done in this field Primarily the transactions in AS/RS can be divided in two parts a) single command cycle, and b) dual command cycle

In single command cycle, either storage or retrieval of items can be performed in a tour of the equipment In this case the storage cycle time is equal to the sum of the times for picking up the load at the I/O station at the end of the aisle, traveling to the storage location, depositing the load in the rack, and returning to the I/O station, retrieval time is also similarly computed

In dual command cycle, both storage and retrievals of the items are made in the same tour of S/R machine In this case cycle time is the sum of pick-up time plus travel time to storage location plus unload time plus travel time to retrieval point plus load time plus return time to I/O station plus unload time

Han et al [1987] have worked on throughput improvement by retrieval sequencing in conventional unit load automated storage/retrieval system, when several retrieval requests are available and dual command cycles are performed Taking first-come-first-serve(FCFS) as the reference sequencing rule, the potential for improvement is identified A "nearest - neighbour" sequencing rule is proposed as an alternative As per conclusions made by them the FCFS assumption is reasonable for storage, since most AS/RS are interfaced with a conveyor loop for input and output In this case, there is no capability for changing the sequence of the loads presented for storage However, for retrievals, the FCFS assumption is less compelling

Also sequencing the retrieval requests optimally is a complex problem, because the list of items to retrieve changes through time as old requests are filled and new requests appear For this situation, two alternatives exists 1) select a block of retrievals, sequence the retrievals in the block, and when the block of retrievals has been completed, select another block and so on, 2) resequence the list every time a new request is added, but employ due dates or priorities to ensure that a retrieval at the far end of the aisle is not excessively delayed In this dissertation first method is adopted for all purposes Now, the outcomes of the work of Han et al [1987] can be summarised as follows

1 10 - 15% improvement in throughput can be obtained by reducing the travel - between component of dual command cycle by 30% or more,
2 an equation is given for approximating the mean dual command cycle time by using a nearest neighbour sequencing heuristics,
3 the nearest - neighbour heuristics obtains average throughput within 5 - 8% of the maximum possible average through

On the other hand in the work of Chaime[1991], an attempt has been made to take advantage of shorter cycle time, but to avoid the considered consequences of the block

sequencing, the nearest neighbour policy was implemented dynamically as a dispatching rule He also showed that the nearest neighbour method is easy to implement and require minor computing resources

1.2.6 Travel Time and Interleaving Time

Generally the travel time models of automated storage/retrieval system assume the average uniform velocity, ignoring the operating characteristics of the storage/retrieval machine such as the acceleration/deceleration rate and the maximum velocity These may result in a design which will be far from optimal To get rid of these, Hwang and Lee[1990] have used randomized assignment policy and have determined travel time for both single and dual command cycle, taking above factors in consideration Specifications of a S/R machine for commercial usage usually include the maximum horizontal and vertical velocities, acceleration and deceleration rate and pick-up and deposit time For these factors, Hwang and Lee[1990] have made assumptions like S/R machine travels simultaneously in the horizontal and vertical directions and the horizontal maximum velocity is not smaller than the vertical maximum velocity, and any point within the pick face is equally likely to be selected for storage or retrieval

The other important parameter in the warehouse operation is interleaving time Enyan and Rosenblatt[1993] have applied the nearest neighbour policy to a class based warehouse in order to decrease the interleaving time Earlier Gravs et al [1977] extended the class-based model of Rosenblatt and Enyan to dual command case and had noticed that the interleaving time comprised about 30% of the total cycle time However, they also showed that an attempt to decrease the interleaving time by changing the shapes of the regions may result in increasing the entire cycle time In the method presented by Enyan and Rosenblatt[1993], a procedure has been developed, which decreases both the interleaving time and the entire cycle time

In their work, they have used the nearest- neighbour policy of Han et al as well as the policy called "dominated areas" or "no - cost zone" According to this policy a retrieval location is selected which belongs to the area dominated by the storage locations Using this approach, the cycle time becomes the twice of one way travel time and no interleaving time is incurred By eliminating the interleaving time component, it was expected that this policy would minimize the cycle time But in the various test conducted, they also found that the cycle time increases with the time the system is operative, and even exceeds the cycle time obtained under the nearest neighbour policy

1.3 Present Work

The research done by Agarwal K , Jain R K [1995] and other relevant works are enough to give insight into the warehouse problem Hence as stated earlier, the work in this dissertation mainly deals with the dual command cycle case in a automated warehousing system In chapter II, we will describe the problem situation and will correlate the work in this dissertation with the earlier works Then we will deal with the synopsis of the classes of the problem and their practical applicability The assumptions made during the formulation and their relevance will also be dealt in the same chapter

The strategy followed here is that we will start with most relaxed case and will go on imposing the constraints as we proceed In chapter III, we will deal with the single bin case, when the equipment used can handle only one bin i e , in a trip of the equipment, it can store only one item and can also retrieve only one The exact methods as well as different heuristics for different cases will also be dealt there The comparisons of the different heuristics among themselves and with the exact methods are also done in the same chapter

In chapter IV, we will deal with the trolley kind of situation, when the equipment can handle more than one bin The heuristics as well as exact mathematical formulations have

been done by keeping in view the capacity of the trolley The assumptions made and the variation of the formulations with the other established methods will be the main features of this chapter

In chapter V, we will deal with the conclusion of our research work, their relevance and will present the summary of the suggestions We will also suggest the logical extension of the problem analyzed for further research in the same chapter

CHAPTER 2

PROBLEM ANALYSIS AND CLASSIFICATION

2.1 Problem Environment

As stated earlier, the work in this dissertation is in continuation of the works done by Bansal R , Agarwal K , and Jain R K [1995] This work primarily deals with the dual command case of the warehouse having a computerised storage/retrieval warehouse system

The design of warehouse may be divided into three parts The first one deals with the selection of location of warehouse with respect to the users of the items in warehouse and the base area of the warehouse, so that an efficient distribution channel can be established As described in different research, Francis' approach of (0-1) integer programming formulation with p - median can be most suitably applied at this stage

At second stage, the internal design of the warehouse is of primary concern The decisions at this stage are also called operational policy decision of the warehouse Here the decisions are made about the configuration of the warehouse, the number and type of aisles, the type and size of racks, positions of input and output docks, the type and number of S/R machines etc As discussed in the chapter I, a lot of work has been done for the decision making at this stage, which can be directly linked with the works in this dissertation As for example, we have taken here the cases for two types of S/R machines, one in which the equipment can carry single bin, and the other in which it can carry more than one bins, but the machine will have specified fixed capacity

The specifications of the S/R machine plays a vital role in determining the travel time and hence the cost matrix in the warehouse Sufficient works has been done on this also, as discussed in previous chapter For purposes of this dissertation, generally we require cost of travel between two identified locations, which can be computed using any of the appropriate cost function

As discussed, each warehouse will have an input dock and an output dock, which may be at same location or at the different location In some cases there may be multiple input/output docks Here we have considered separately the cost of travel to any location from input dock and output dock, and in case there are multiple docks, the cost from closest input/output dock is taken However the cost of travel by equipment in return journey from output dock to input dock, if they are at different locations, is ignored Also we have assumed here that a S/R machine can't break its journey in midway of the racks, i e the machine will certainly complete its tour up to the output dock

We have considered the inter-aisle movement which will also include vertical movement of the equipment in the intra-aisle tour, which may be any of the following types

- the equipment has to be moved to the ground position and then again to appropriate height after attaining required horizontal movement
- the equipment can be moved horizontally without changing the vertical position and hence vertical distance moved will the difference of the vertical locations of the two points
- in some cases the pick-up arm can be moved across the rack along the shortest path

The distance measurement for different cases can be done by varying the coefficient's value as discussed in the appendix[A] and can be used for computing the cost between the spatial locations

The third stage of the warehouse design is the functional policy decision making At this stage the decisions are made about day-to-day functioning of the warehouse, like, where to store an item, from where to retrieve an item, which equipment should be used etc The work in this dissertation is mainly concerned with the decision making at this stage As discussed earlier, the policy adopted for storage may be either dedicated storage policy or it may be shared storage policy In this dissertation only shared storage policy is considered for the formulation of problem, but relaxation has been incorporated so that items can be grouped in such a way that for each group of items, specific aisles can be allocated In this the inter-aisle movement can be restricted and hence we are able here to adopt the methodologies, which are independent of the aisle's interaction For grouping of the items, the class-based on intra-aisle structure can be utilised A lot of work has been done on this topic, and any of them can be referred for more details The other approach for grouping of the items is as discussed by Rosenwein[1994] He has formulated the clustering problem as p-median 0-1 integer program which can be solved very efficiently

Further dual command cycle is the combination of the stock assignment problem and the order picking problem, in which both the functions of storing the item and retrieving the same are done in a single tour of the S/R machine These two problems can be described separately as follows

2.1.1 The Stock Assignment Problem

This problem is essentially, is to decide where to store the item in the warehouse, so that the retrieval of these items can be done optimally and the storage cost is also minimized Thus both storage cost and retrieval cost are to be minimized at the time of storage itself Whenever a storage request comes to the warehouse, the system optimally locate the items among the locations available for storing those items Optimality criterion will take

into account the cost of travelling to the storage location and a shadow cost related to the retrieval cost while storing the item or it can be the minimization of the retrieval cost of the items from this location Another case is when retrieval is done in batches where the optimality criterion may be to store the items near to the items with which it is frequently retrieved so as to minimize the tour length of the equipment while retrieving the order Or the objective may be a combination of both of the above objectives

Whenever a storage request comes to the warehouse and their are various types of equipment which can handle the request, in addition to decide the locations, where the items are to be stored, the information system should be capable of selecting the best equipment for serving each location which has been chosen for storing the item

2.1.2 The Order Picking Problem

The order picking problem is the complementary of the stock assignment problem discussed above Whenever a retrieval request comes to the warehouse, the system should decide an optimal sequence for picking the items listed in the request from their respective storage locations Here also there may be several cases depending upon the type of the request menu Retrieval request can be of a single bin of an item or it can be a menu comprising of several bins of an item The objective criterion generally will be the minimization of the retrieval cost Again when many type of equipment are there then it is to be decided which equipment will retrieve which item from what location When equipment can handle more than one bin at a time, the locations and items which should be served in a particular tour have to be decided When more than one retrieval requests are there, then keeping the due date constraints in mind, pick up orders can be formed for actual orders i e , items from two or more actual orders can be clustered together to form a pick-up order to be retrieved in a single tour

2 2 Problem Classification

Following considerations apply to classify the problem

I As discussed earlier, the equipment used in the warehouse may be of the many kind The alternatives considered are to use single bin carrying equipment or to use trolley attached truck with a fixed capacity

II The order in which items can be stored and retrieved, if they are in cluster This implies that if we are using the single bin carrying equipment and if we have more than one items to store, then the order of the items for storage Same consideration is applicable in case of retrievals also

III The dependency of the type of items in decision making, i e , whether at the time of selection of the storing locations, the identity of the items should be considered or not For all practical purposes, the methodologies developed are assumed to be item independent However, the cost related to the item has been taken into consideration as mentioned in the appendix A

IV As is evident, in dual command case, the storage and retrievals of different items are done in the single tour of the S/R machine Now the cases may be different when we can reuse the just vacated location for further storage and when we don't do so This is applicable only in the case when the storage and retrievals are done for a group of items

Depending on the bases considerations discussed above, the dual command cycle may be broadly classified of three types

I. **Single item storage/retrieval case :** In this case, we have only one item to store as well as only one item to retrieve Hence we are bothered here only for the optimality of the single tour of the equipment This is applicable to the situation when after each tour the list of storing and retrieving items is updated

II. **Single bin carrying S/R machine case :** In this case we have a menu of items which are to be stored and similarly we have a menu of retrieving items The decision making here is the selection of the location where the item can be stored as well as selection of the item and location from where it has to be retrieved

III. **Multi-bin carrying S/R machine case :** In this a trolley is attached with the truck and hence the equipment is capable of carrying more than one bin at a time Thus case is considered with the assumption that the storage and retrieval can be done in any order, depending on the capacity available with the equipment

Different approaches to solve above classes of problem are discussed in more details in chapter III and IV

2.3 Assumptions

In this section we list all the assumptions that have been made in formulating the problem Specific assumptions, if any, appear along with the respective cases

- The warehouse in consideration is a rectangular warehouse with an input dock and an output dock
- Storage racks have two dimensional structure with several rows and columns partitions Unless otherwise specified, uniform partitioning has been assumed i e , all storage locations are similar and any item can be stored anywhere
- Bin sizes are standardized and are of unit sizes
- Each bin holds only one type of item at a time
- Storage and retrieval requests are for the entire contents of the bin
- Equipment selection consideration do not apply to our analysis However, in all cases it has been assumed that for a given tour, the best possible choice has already been made

- Storage requests are served on first-come-first-serve(FCFS) basis Once the order size for storing and pick-up has been fixed, no updating is done unless all assignments are made
- Retrieval requests are not served on any basis, the only criteria considered to select the retrieving item is the total cost of the retrieval
- Equipment scheduling is not considered
- It is assumed, there will not be common item in the storing menu and in retrieving menu If there are any such items, they will be matched out prior to the assignment of the other items, and directly transferred to output dock

CHAPTER 3

SINGLE BIN STORAGE/RETRIEVAL SYSTEM

In this chapter we are going to deal with the case of storage and retrievals of items, when the S/R machine can carry only one bin at a time The discussions here can be broadly classified in two parts In the first part we shall consider the case when the number of items to store and to retrieve are only one each, and the second part deals with the situation when the numbers of items to store as well as the number of items to retrieve are more than one, i e, the S/R machine will have to make more than one tour for completion of full assignment

However, the only objective function, which we have considered in all the cases is to minimize the traveling cost The travel cost between two spatial locations are known apriori For any location of a rack, in the warehouse system, there are three types of cost associated with it Among these, one is the cost of the travel from input dock, second is the cost of the travel to the output dock and third is the inter-location cost Depending on the type and movement of the S/R machine, these costs can be determined in a number of ways A generalised equation for the distance measurement is developed by Agarwal K [1995] and Bansal R [1995] for inter-location distances, which can be used to measure inter-location costs[appendix A]

Now if we take 'a_i' as the cost of travel to location 'i' from input dock, 'b_j' as the cost of travel to location 'j' from output dock and 'c_{ij}' as the cost of travel between the locations 'i' and 'j', then for the case considered, the cost of travel incurred by equipment in storing one item at location 'i' and retrieving another item at location 'j' is

$$c_{ij}' = a_i + c_{ij} + b_j$$

As for the case here, equipment can carry only one item at a time, hence for every storage at location 'i' and for every retrieval at location 'j', there will be a unique cost 'c_{ij}" associated with the above assignment Hence in the objective functions for different cases only this cost is considered for the storage of an item at location 'i' and retrieval of another item at location 'j' Now as this cost 'c_{ij}" also incorporates the cost of travel from input dock and output dock apart from the cost of travel between the locations 'i' and 'j', hence the cost 'c_{ij}" is taken different from cost 'c_{ji}"

The input to the problem is assumed to be the number of empty locations, name and number of items to store, the name and number of items to retrieve, the locations from where these items can be retrieved, and all the travelling cost associated with these locations We have also assumed here that an item can be stored at any place irrespective of where the other bins of same item are stored, i e , it is not necessary that all bins of an item should be stored in adjacent locations Hence the cost of travel for different existing locations of an item are dependent on the locations

3.1 Single item storage/retrieval

We have considered two approaches to solve this problem, among which one is optimal and another is heuristic

Let,

m number of locations, where an item can be stored,

n number of locations, from an item can be retrieved,

C_{ij} value of the objective function if the storage is done at location 'i' and retrieval is done at location 'j',

1, 2, , m available empty locations,

m+1, m+2, ,m+n available retrieving locations,

then the problem can be represented by following network

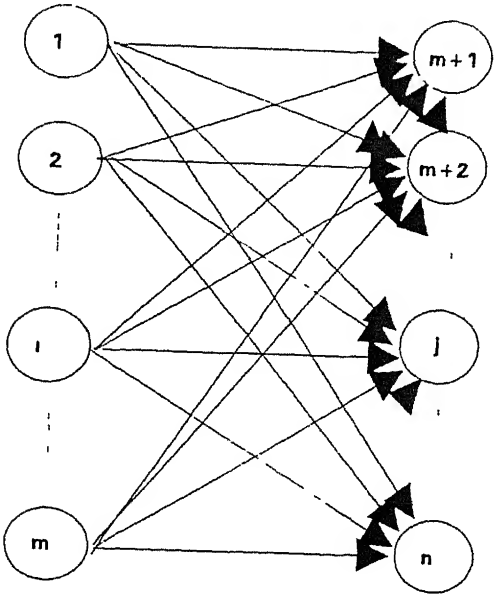


figure 3.1

In this network every node in storing side is connected with every node in retrieving side Different methods for solving this problem are described as follows

3.1.1 Exact Method [SSP]

In this method all the possible paths for carrying out the assignment are explored and optimum among them is selected, i e , in the above network we will select the arc which will have minimum cost associated with it As there are m nodes in storing side and n nodes in retrieving side, the number of arcs is m * n

3.1.2 Heuristic Method [SNN]

For a single bin storage/retrieval system above method is efficient However we will discuss the approximate method, as in subsequent discussions it will help in designing heuristics for more difficult problem and for speedy on-line implementation `This approach is quite similar to nearest-neighbour approach as discussed in previous chapter In this case step by step decision is taken for the path of the equipment At first

we select the storing location which is nearest to the input dock and then select the retrieving location which is at nearest to the storing location selected earlier As evident, the number of arcs considered here is of the order of O(m + n) In this approach, the distance of the retrieving location from output dock is not taken under consideration

3.2 Multi - item Storage/Retrieval

In this case, there is a menu of the items, which are to be stored and similarly there is a menu of the items which are to be retrieved The decisions are made about the determination of the tours of the S/R machine, so that required number of storage and retrieval can be made As the S/R machine is capable of handling one item at a time, machine will have to make more than one tour to complete the assignment Also in all cases the equipment will first execute the storage work and then will go for the retrieval of an item Further in case the number of retrievals required are different from the number of storage, some of the trips will be only for the storage or retrieval In all such cases after simultaneous retrieval and storage, the additional storage and retrieval will be done on the least cost basis Methods which we are discussing generally will ignore this part of the assignment, as it can be done trivially

We have further simplified the problem by identifying the representative location for an item which should be retrieved, with each storage location Thus with each storage location, a specific retrieval location for each item is pre identified We already know the fact that a unique cost is associated with every pair of storage and retrieval location Consider an example, in which there are 'm' locations from which an item 'j' can be retrieved Let 'i' be one among the empty locations, where an item can be stored Now we will have one unique cost 'C_{ik}', when we make storage at location 'i' and retrieval at location 'k" At the time of decision making only one of the m locations will get selected

for the retrieval of item 'j' Hence if we select apriori the location 'k' among the existing locations, for which the cost is minimum among the m locations from location 'i', then we can define cost $C_{ij} = C_{ik}$ for item 'j' Then during the decision making whenever the path 'i → j' will be selected, it will refer to the actual path 'i → k', thus we will state the location 'k' as representative location for item 'j' associated with the storing location 'i' Similarly we can decide apriori about the representative location for each of the retrieving location and with each of the empty locations Thus node 'j' will represent for each storing node 'i', a set of representative locations As can be seen in the later sections, we can reduce the number of nodes at different stages significantly by this approach during decision making Also complexities of the problem will get reduced, as instead of dealing with total number of locations where the retrieving items exist, we have to deal only with the number of retrieving items

Further this problem can be sub-divided into two cases, one in which the just vacated location can't be used for the further storage purposes, and the other in which the just vacated location is considered for further storage However in none of the cases we are going to deal with the retrieval of just stored items as in such case, the equipment can bypass the warehouse storage locations altogether and directly transfer it to output dock The solution approaches for these two cases of the problem are different and will be analysed in more detail in the remaining sections of this chapter In all the cases we will deal with the exact methods, if any, first and then will analyse the heuristic methods The analysis of the results for the same set of problems for exact and heuristic methods is discussed in section 3 3

The input to the problem is assumed to be the menu of items to be stored, menu of items to be retrieved, locations available for storage, locations available for the retrieval of the items, and all the costs associated with these locations

3.2.1 Storage/Retrieval Without the Use of Vacated Locations

In this case we consider a menu of storage and retrieval items such that, no item which is required to be stored is on the retrieval menu If there are such items a direct transfer will be made, and problem reduced accordingly Further the locations which are made available by retrieval of items for this menu can not be used for the storage of the remaining items in the menu The implications of this limitation is that information update about the available storage location is done only after all the items on the menu have been stored/retrieved

3.2.1.1 Exact Methods

The problem can be efficiently formulated as well as can be solved optimally We have developed here two type of formulation, one which we call network formulation/integer programming formulation and the other is dynamic programming formulation

3.2.1.1.1 Network/integer Programming Formulation

The problem here can be easily formulated as the minimum cost flow problem First we shall describe the standard network notations and then later will describe the interpretation of these notation for the problem under consideration

l_{ij} lower capacity bound on the flow over arc(i-j),

u_{ij} upper capacity bound on the flow over arc(i-j),

p_k net flow at node 'k',

C_{ij} cost per unit flow from node 'i' to node 'j'

X_{ij} flow from node 'i' to node 'j'

For the problem considered, the network and representations of variables can be described as follows

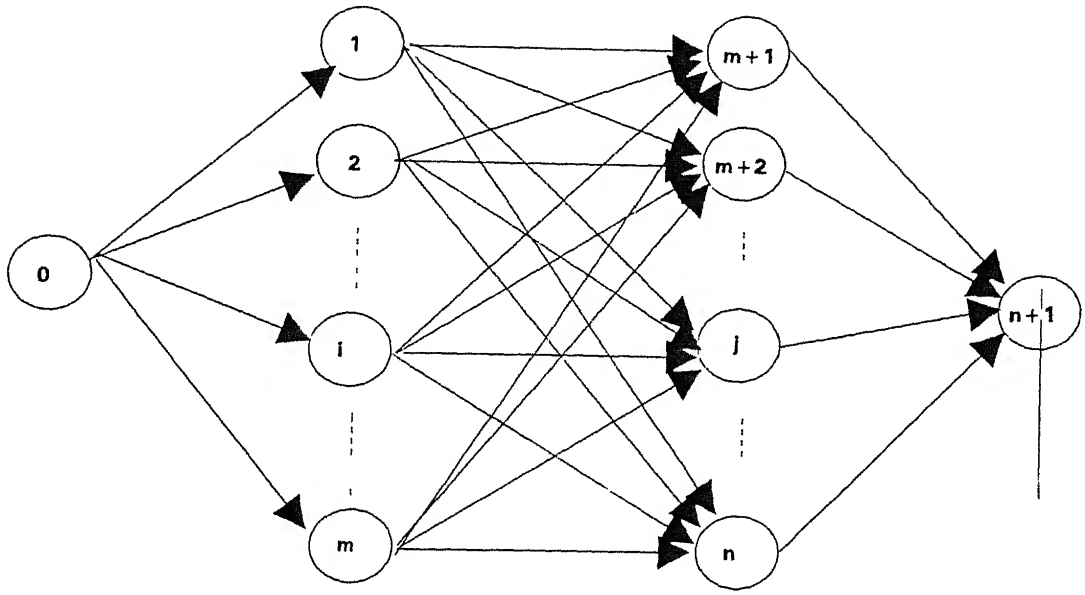


figure 3 2

Let there be m storage location required to retrieve k items Firstly we will identify with each storage location a representative location for each item In the network we will have total of n = m + k nodes, corresponding to m storage locations and k items Node '0' is of symbolic importance only, which is included here to just represent the number of assignments to be made, hence p_0 is equal to the desired number of the tour of the S/R machine Similarly node 'n+1' is sink node which just ensure that required number of retrievals has been made, for this node p_{n+1} is equal to the minus of the p_0 Also in the network nodes '1' to 'm' represents available storage locations and nodes 'm+1' to 'n' represents the locations for retrieval(note these nodes are just representative locations for a specific item and will have different interpretation with the selection of different storage location 'i') However as stated earlier for each item to be retrieved there is a unique retrieval location associated with a specific storage locations

Let,

$$S = \{i, i = 1, 2, \quad , m\}$$

$R = \{j, j = m+1, m+2, \quad ,n\}$

Hence S is the set of storage locations and R is the set of retrieving items As shown in the network, every element of set S is connected to every member of set R by directed arcs with head in set S and tail in set R This signifies that the equipment can store any of the storing items and then can retrieve any of the retrieving items Also every arc from the elements of set S to elements of set R represent that the equipment will travel from input dock, will store a bin of an item at location 'i'($i \in S$) and retrieve a bin of item 'j'($j \in R$) from the corresponding representative location and then will go to the output dock Hence the cost 'C_{ij}' associated with the arc 'i-j' incorporates all the cost associated in completing above described tour of the equipment

The representation of other notations are as follows,

$X_{ij} = 1$, if storage is made at location 'i' and retrieval of item 'j' is made (where $i \in S$ and $j \in R$)

$= 0$, otherwise,

$p_k = 0$, for all $k \in (S \cup R)$,

$p_0 =$ minimum(a, b),

where a = number of items to be stored,

b = number of items to be retrieved,

$p_{n+1} = - p_0$,

$l_{ij} = 0$, for all arcs in the network,

$u_{ij} = 1$, for all arcs in the network

Now the mathematical presentation of the problem can be done as follows

Minimize $\sum_{i \in S} \sum_{i \in R} C_{ij} X_{ij}$ **...(3.1)**

Subject to,

$$\sum_{i \in S} X_{oi} = \min(a, b) \quad \text{...(3.2)}$$

$$\sum_{k \in R} X_{ik} - X_{0i} = 0, \qquad i \in S \qquad \text{...(3.3)}$$

$$X_{j(n+1)} - \sum_{k \in S} X_{kj} = 0, \quad j \in R \qquad \text{...(3.4)}$$

$$-\sum_{k \in R} X_{k(n+1)} = -\min(a, b) \qquad \text{...(3.5)}$$

$$0 \leq X_{ij} \leq 1 \qquad i \in S, j \in R \qquad \text{...(3.6)}$$

$$\text{integers} : X_{ij} \quad i \in S, j \in R \qquad \text{...(3.7)}$$

In the above formulation, it has been assumed that no cost is associated with the arcs from node '0' to i(i ∈S), and similarly no cost is associated with the arcs from nodes j(j ∈ R) to the node 'n+1' Also the number of nodes here is n + 2 and the number of arcs are of the order of O(m * k) This formulation of the problem can be solved optimally by any of the minimum cost flow algorithms

We have used the 'NETOPT' package available with 'CPLEX' to solve the above network formulation of the problem This method of network optimization is considered to be hundred time faster than the other optimization technique for similar LP formulation In sections 3 3 the results obtained by this technique are analysed

3.2.1.1.2 Dynamic Programming Formulation

An alternative solution methodology is to use Dynamic Programming While Dynamic Programming in this case is inefficient compared to approach mentioned above, it will help us in designing heuristics in more complex cases

Dynamic programming also known as "multi stage programming"[Taha H A 1992], is related to branch and bound in the sense that it performs an intelligent enumeration of all the feasible points of a problem, but it does go in a different way It basically dictates how a properly decomposed problem may be solved in stages(rather than as one entity) through the use of recursive computations

In light of the problem at hand, various terms for dynamic programming have been defined next

For our problem, the locations where an item can be stored will form various stages of dynamic programming '0' stage will represent initial stage Thus there will be (m + 1) stages in this problem Decisions at any of the stage will correspond to either no storage at the next location or storage of an item and retrieval of an item not retrieved so far A state is normally defined to reflect the change in status, with respect to decision, of the constraints that bind all the stages together For this problem, a state will refer to the subset of the items which have already been retrieved including the item being retrieved at that stage If at any stage no item has been retrieved it will be represented as '0' in the subset Thus a state at k^{th} stage is a k-tuple subset, consisting of the items which have been retrieved so far (including at that stage) and replications of '0' for balance number, i e difference between k and number of items retrieved Therefore, the states at any stage represent all possible combinations of retrieving locations, which can decide the tours of the equipment up to that stage Let us take a practical case, in which we have to take decisions at third stage Let number of empty locations where the storage can be made is seven, and the number of items to be retrieved is three Hence the possible decisions at this stage are either not to store any item at this stage, or to store one item and to retrieve one particular item among the items, for which retrieval is to be made At a stage, in a particular state only those decisions are feasible which correspond to either no storage (0)

or a storage and retrieval of an item, which have not been retrieved so far i e at stage 'j' and state X_i^j, $d^j(X_i^j) \neq k \in \{X_i^j - 0\}$

Further from stage 'j - 1' state X_k^{j-1} and decision $d^{j-1}(X_k^{j-1})$, the system state at the j^{th} stage will be,

$$X_i^j \equiv X_k^{j-1} \cup d^{j-1}(X_k^{j-1}) \qquad (3\ 8)$$

Let us take a state at third stage, in which the retrievals of items a, b, c are to be done simultaneous to the storage of items at locations 1, 2, and 3 Now we can only correlate this state with the states at second stage, at which we have already made decisions about the retrievals of the two of the items i e , this can be related with states (a, b), (b, c), and (a, b) at the second stage Now the decisions associated with the previous stage is to select item a, or item b, or item c, respectively from the previous states (b, c), (a, c) and (a, b), which will lead to the state (a,b,c) If decision not to store any item at a location has been done, it will get reflected as '0' in the subsequent states

Thus,

decision $d^j(X_i^j)$ → decision at state 'X_i^j' for stage 'j' is that which (if any) item should be retrieved when the equipment will make tour to store an item at location 'j + 1' $d^j(X_i^j) = 0$ indicates that no storage is being made at next stage

Further, $d^j(X_i^j) \neq k \in \{X_i^j - 0\}$

revenue function (R^j $d^j(X_i^j)$) → it is the travel cost of the equipment to location 'j + 1' from 'j' and subsequently to the retrieving item locations corresponding to the decision $d^j(X_k^j)$ In case $d^j(X_k^j) = 0$, the cost $R^j(d^j(X_i^j)) = 0$

state X_i^j → is a k-tuple, each element of which represent an item which is retrieved upto that stage, and replication of '0' corresponding to number of stages, where no item has been stored It is an unordered set Further, $X_1^0 \equiv \{0\}$

optimal return function($F^j(X_i^j)$) → given the state 'X_i^j', it gives the optimal return upto that stage Recursive equation for this is stated below

The state relationship for binding the successive stages together can be given as follows

$$X_i^j \equiv X_k^{j-1} \cup d^{j-1}(X_k^{j-1}) \qquad \text{...(3.8)}$$

Also the dynamic programming gives us the following recursive relationship

$$F^0(0) = 0 \qquad \text{...(3.9)}$$

and,

$$F^j(X_i^j) = \min\{R^{j-1}(d^{j-1}(X_k^{j-1})) + F^{j-1}(X_k^{j-1})\}; \text{ for all } j \geq 1 \text{ ...(3.10)}$$
$$\text{for all } X_k^{j-1} \text{ such that } X_i^j \equiv X_k^{j-1} \cup d^{j-1}(X_k^{j-1})$$

In this way we can get the optimal solution for the problem by forward decision process of the dynamic programming By tracing back the optimal solution from end stage to first stage, we can get the optimal tours of the equipment for the assignments made

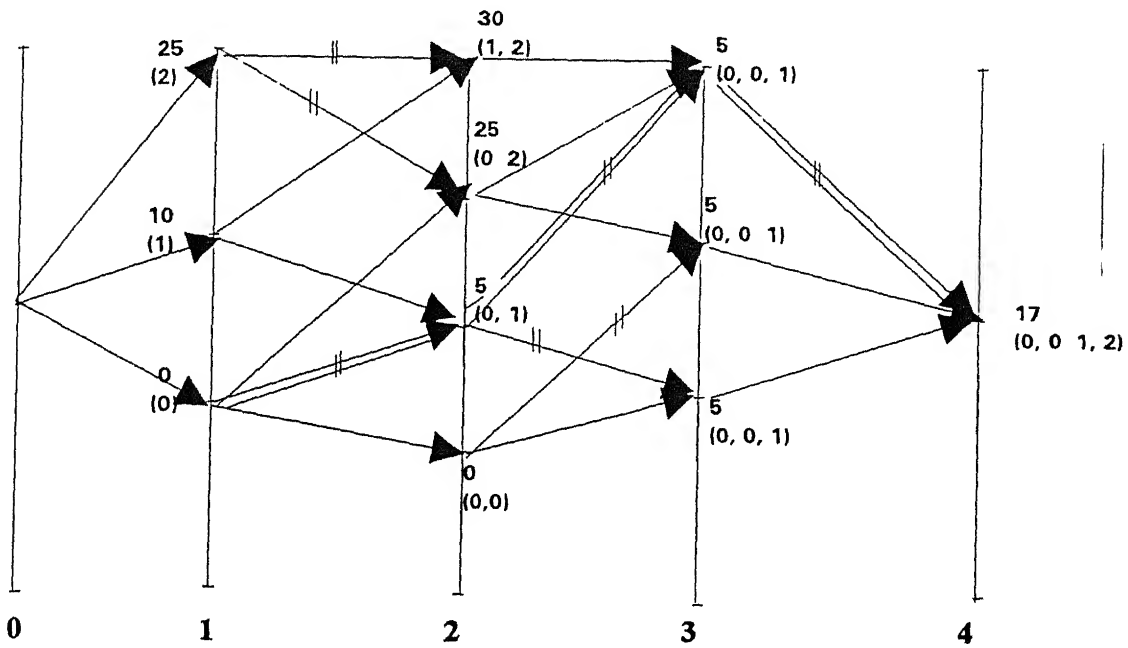
The maximum number of decision arcs at a stage may be of the order of $O(2^n)$, where n is the number of retrieving items An example is shown in figure 3 3, for the clear understanding of the approach As will be discussed in later sections, this methodology gives optimal result, but the time of execution increases exponentially with the increase in the number of stages and the number of items to be retrieved

EXAMPLE FOR DYNAMIC PROGRAMMING

Input:-

no of empty locations = 4
no of items to be retrieved = 2
no of items to be stored = 3
cost matrix

storing locs	1	2	3	4
retr locs				
1	10	5	7	15
2	25	15	12	14



The elements in bracket represents at states at a stage and the number above it represents the cost of the decision The optimal path is represented by the double line
Which can be interpreted as,
equip tour 1 storage at location 2 & retrieval of item 1,
equip. tour 2 storage at location 3 & retrieval of item 2

The optimal value of objective function is 17

3.2.1.2 Heuristic Approach

For the problem, we have developed two heuristics, and they are found to be very efficient as well as give solutions close to the optimal solution These methods are analysed in detail in subsequent sections

3.2.1.2.1 Heuristic I (HDP)

This methodology is similar to the dynamic programming approach In this case number of stages will be equal to the number of items to be stored subject to, a maximum of number of items to be retrieved In case there are more items to be stored than the number of items to be retrieved, excess items will be directly stored after the storage locations and retrieval locations are matched

A state at a stage represents the combination of a storage location with a retrieval item However as for each item a representative location(closest to a storage location) can be selected apriori, each such combination reflects a storage-retrieval location combination If m is the number of empty locations and n is the number of items to be retrieved, then we can make the tour of the equipment in m * n ways, and it is the number of states at a stage Also with each node(i, j) a unique cost is associated, which is proportional to the distance travelled by the equipment from input dock to the location 'i', plus the distance of the location 'k'(representative location for item 'j', when storage is to be made at location 'i') from location 'i' plus the distance from location 'k' to the output dock let this cost be C_{ij} At each stage for each state the minimum cost of reaching upto that state will be computed as follow

For each state(i, j) let,

$I^k_{(i,j)}$ = { list of al the storing locations selected so far including at this stage 'k'}

$J^k_{(i,j)}$ = {list of all the retrieving items selected so far including this stage 'k'}

then

$$C^k_{(i,j)} = c_{ij} + \min_{\substack{i \notin I^{k-1}_{(l,m)} \\ j \notin J^{k-1}_{(l,m)}}} C^{k-1}_{(l,m)}$$

This implies that for the state (i, j) at stage 'k', from each state (l, m) of the previous stage 'k - 1', such that location i and j are not yet selected upto that state, the sum of the cost of the tour (i, j) and the minimum cost upto the state (l, m) is taken as the minimum of all these computed costs

The methodology can be described stepwise as follows

Step I Make the list of all possible combinations of the storing locations and retrieving items (i e states at a stage), let us call it list 1 The cost associated with each member of the list should also be computed at this stage

Step II For each stage maintain with each state, the cost to reach upto that state, as well as the list of the items, storage locations selected so far

The cost function for state (i, j) at stage 'k' can be given as

$$C^k_{(i,j)} = c_{ij} + \min_{\substack{i \notin I^{k-1}_{(l,m)} \\ j \notin J^{k-1}_{(l,m)}}} C^{k-1}_{(l,m)}$$

Update the list for each of the states at that stage

Step III Repeat the step II for required number of stages ,i e , minimum of number of items to be stored and number of items to be retrieved

Step IV Select the state at final stage, which will have minimum value of the objective function

At any stage, if it is not possible to make assignment for a state, delete that node from the list, and no further consideration will be given for this state in subsequent stages This heuristic is tested for a large number of problems, the result's analysis is done at the end of this chapter

3.2.1.2.2 Heuristic 2 (SPP)

This heuristic is much simpler than the heuristic1, which has been described earlier The heuristic is an iterative procedure where each time, among the available storage and retrieval locations the shortest path tour is selected The selected locations are removed from the storage and retrieval list and the processes are repeated The heuristic can be described stepwise as follows

Step I Make all combinations of the empty locations, available for storage, and the retrieving items, let us call them nodes Compute the cost associated with each node Here also as earlier case, the cost C_{ij} associated with a node i-j, incorporates in making the tour of the equipment from input dock to the location 'i', from this location to location 'k' (representative for the item 'j', when the storage is made at location 'i' and retrieval is made of the item 'j')

Step II Select the node (i, j) for which the cost C_{ij} is minimum

Step III Delete all the nodes from the list which contain either the locations of the selected node

Step IV Repeat the step II and III, until required number of decisions are made, this number is equal to the minimum of the number of items to be stored and number of items to be retrieved

Step V The decisions made at different stages decide about the tours of the equipment Each of the decisions represents one tour of the equipment Let n is the number of

decisions made, then value of the objective function will be the sum of C_{ij}s for n decisions made at various stages

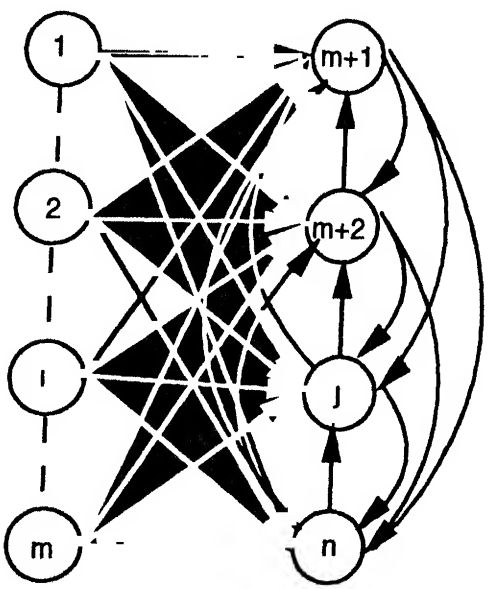
In this approach maximum number of decision variables is m * n at first stage and then it goes on decreasing in succeeding stages

3.2.2 Storage/Retrieval, When the Vacated Locations are Permitted for Further Storage

In this case, we consider the problem when a location from which a retrieval is made, is assumed to be used as a storage location subsequently Thus the list of locations available for storage will dynamically get modified and will include location just vacated by a retrieval However the items stored at any stage, will not be considered for the retrieval Except cost matrix, the input here is assumed to be the same as prior cases

At first, we will discuss the exact integer programming formulation of the problem and then will discuss heuristics for the same

3.2.2.1 Integer Programming Formulation



The network shown above is for the case, when we have m empty locations where item can be stored directly and have 'n-m' locations from where retrievals can be done We have made the assumption that the sets of the retrieval locations where an item exists and the sets of storage locations where storage can be made are given as input to the problem These sets of locations and items can be represented as follows

Let,

$r_1, r_2, \quad , r_b$ are different items, which are to be retrieved

$R_{r_i} = \{k, k \text{ is the location where item } 'r_i' \text{ exists}\}$ for $i = 1, 2, \quad , b$

$R = \{l, l \in R_{r_i} \text{ for all } i = 1, 2, \quad , b\}$

$S = \{1, 1 \text{ is the initial storage location}\}$

$I_i = \{p, \text{ where } p \in R_{r_k}, \text{ for all } k \text{ and } i \notin R_{r_k}\}$ for $i = 1, 2, 3, \quad , n$

$J_j = \{q, q = i, \text{if } j \in I_i \text{ for all } i = 1, 2, \quad , n, i \neq j\}$ for $j = m + 1, m + 2, \quad , n$

Hence above representation implies that S is the set of all storing locations, R is the set of original retrieving locations, R_{r_i} is the locations where a retrieving item r_i exists Similarly I_i is the set of locations from where the items will be retrieved if the storage will be made at location 'i', and this prevents the arcs in between the locations where same item exists J_j is the set of locations 'i' for which 'j' is the member of I_i Therefore in the network shown above, for a node 'i' all outgoing arcs should be to the nodes which are member of set I_i as well as for a node 'j' all incoming arcs should be from the nodes which are member of the set J_j

Here we formulate the problem as Integer Programming problem Different notations in addition to above can be represented as follows

For any arc $i \rightarrow j$,

$X_{ij}^t = 1$, if storage is done at location 'i' and retrieval is done at location 'j' at step t,
$i \in (S \cup R)$ and $j \in R$

= 0, otherwise,

where, S = {i, i = 1,2,3, , m}

R = {j, j = m + 1, m + 2, ,n}

C_{ij} = cost incurred in the travelling of equipment, when an item is stored at location 'i' and another item is retrieved from location 'j'

a = number of items to be stored,

b = number of items to be retrieved,

M = min(a, b), it indicates the number of tour of equipment to be decided by methodology

The basic difference in this case from earlier one is that here the arc $i \rightarrow j$($i \geq m + 1$ and $j \geq m$) represents that location 'i' has been vacated by some earlier tour of the equipment, and hence an item can be stored at location 'i' and retrieval of another item can be done from location 'j', in a tour of the equipment

The IP formulation of the problem can be given as follows

Minimize $$\sum_{t=1}^{M}\sum_{i=1}^{n}\sum_{j\in I_i} C_{ij} X_{ij}^{t} \qquad (3\ 11)$$

Subject to,

$$\sum_{i=1}^{m}\sum_{j\in I_i} X_{ij}^{1} = 1, \qquad (3\ 12)$$

$$\sum_{t=1}^{M}\sum_{j\in I_i} X_{ij}^{t} \leq 1, \qquad i \in S \qquad (3\ 13)$$

$$\sum_{j\in R_p}\sum_{i\in J_j}\sum_{t=1}^{M} X_{ij}^{t} \leq 1, \qquad p = 1, 2, \ , b \qquad (3\ 14)$$

$$\sum_{i\in R_p}\sum_{j\in I_i}\sum_{t=1}^{M} X_{ij}^{t} \leq 1, \qquad p = 1, 2, \ , b \qquad (3\ 15)$$

$$\sum_{t\ 1}^{M}\sum_{i=1}^{m}\sum_{j\in I_i} x_{ij}^{t} + \sum_{i=m+1}^{n}\sum_{j\in I_i}\sum_{t=1}^{M} x_{ij}^{t} = M, \qquad (3\ 16)$$

$$\sum_{k=1}^{t-1}\sum_{i\in J_j} X_{ij}^{k} - \sum_{\substack{i\in j_j \\ i>m}} X_{ji}^{t} \geq 0, \quad t = 2, 3, \ ,M, j \in R \qquad (3\ 17)$$

$$\sum_{i=1}^{n}\sum_{j\in I_i} X_{ij}^{t} = 1, \qquad t = 1, 2, \ , M \qquad (3\ 18)$$

$$X_{ij}^{t} = (0, 1), \qquad \text{for all } i \rightarrow j \qquad (3\ 19)$$

In this formulation, besides the constraints put in the formulation of the problem in section 3 2 1 1 1, additional constraints have been put mainly to prevent the formation of cycles, which may lead to infeasible solution Here constraint (3 12) implies that the first decision must be to store an item only at the locations 'i', where $i \in S$, and to retrieve another item from location 'j', where $j \in R$ Constraint (3 13) implies that to a location 'i', $i \in S$, equipment can visit only once, and that also for storing purpose only Constraint (3 14) restricts more than one visit of equipment for the retrieval of an item, whereas constraint (3 15) implies that the location vacated so, should only be used once for storage Constraint (3.16) guarantees that the number of tours decided by the methodology will be equal to the minimum of number of items to be retrieved and number of items to be stored Constraint (3 17) is for the restriction of the cycles, if any It says that there can be an outgoing arc from a node at a stage, if and only if, the same node has been visited earlier for retrieval purpose As the formulation is for sequential kind of decision making, hence this constraint disallow any tour of the equipment, in which it can go for storage at a location , which is not made vacated in earlier stages For the

restriction of number of decisions which can be taken at a stage to one only, constraint (3 18) has been put here

For the computational analysis, we have considered simplified problem We have assumed that the number of locations from where an item can be retrieved is only one Hence in this case, following changes will take place

$n-m = b,$

$I_i = \{j, j = m+1, m+2, m+3, \quad ,n, i \neq j\}$ for $i = 1,2, \quad , n$

$J_j = \{i, i = 1, 2, 3, \quad , n, i \neq j\}$ for $j = m + 1, m +2, \quad , n$

After incorporating these changes, we have run this programme for sufficient number of problems by using 'CPLEX' Results has been discussed in the section 3 3

However as the method is not suitable for large problems, we will further develop some simple heuristics for this problem Here in this dissertation three such techniques has been described, which are simply the modified techniques described in the section 3 2 1

3.2.2.2.1. Heuristic 1 (HVDP)

The approach here is similar to that described for heuristic in section 3 2 1 2 1, with the only difference that on second stage onwards we also consider the possible decisions of storage at just vacated locations, i e at the locations where retrieval have been made in same order This consideration increases the number of states by n * (n - 1), if the number of retrievals to be made in a order is n, and hence the maximum number of states at second stage onward will be (m * n) + (n * (n - 1)), where m is the number of available empty locations

Here only caution required in addition to that in section 3 2 1 2 1, is that at the time of decision making at a stage for making storage at the just retrieved location (say i), it should be assured that the item at 'i' has already been retrieved and a location is not going to be revisited for same purposes in any of the possible path in the network The number

of stages in this case is also same as that in heuristic (HDP) i e equal to the minimum of the number of the items to be retrieved and the number of items to be stored in same order Also the idea of representative location for an retrieving item is equally applied as that in other related cases

3.2.2.2.2. Heuristic 2 (HVSP)

This heuristic is also exactly similar to the heuristic (HSP), described in section 3 2 1 2 2 Here after making a decision at a stage, we not only delete the nodes whose elements are common to either storing location or with the retrieving item or with the both, but we also add the nodes which permit us to make decision of storing at the location, where retrieval is made at the preceding stage, and making retrievals of the items, which are yet to be retrieved Also care should be taken in keeping track of the retrieving items, for which retrieval decisions have already been made Hence the methodology adopted also maintains a list of retrieving items for which decision has already been made, and by matching with this list we can ensure that none of element of none of the state at any stage will have these items As evident, at the second stage the number of states will change by (n -1)(due to the addition of some new alternatives) minus (m + n)(due to removal of some alternatives) However this number goes on decreasing as we proceed to next stages

3.2.2.2.3. Heuristic 3 (HVEDP)

This heuristic is the improved version of the dynamic programming formulation of the problem as discussed in section 3 2 1 1 2 The difference here with the earlier case is that the number of locations where an item can be stored gets increased dynamically, as locations from which retrieval are made are added as and when they become free, thus

number of stages will increase from m to m + n, where m is the original storage locations and n is the number of items to be retrieved and these new stages are added after the m^{th} stage

The state descriptions, possible set of decisions and revenue functions are same as described in the section 3 2 1 1 2, except that for stage m onwards, from a state X_i^J a link can be established to the next stage, only if one of the item retrieved so far is corresponding to the locations at the next stage In other word for stage m onward $d^J(X_i^J)$ = $\{\phi\}$ if X_i^J does not contain item corresponding to original retrieval location at stage J + 1

All other procedures of solving the problem are same as that in section 3 2 1 1 2 However in contrast to the earlier case, the solution varies with the change in order of stages after m^{th} stages This can be further illustrated by taking an example Let us make the stage m + 1, corresponding to the location vacated by item 'j', then the alternative decisions at this stage will be different than the situation when this stage will correspond to the some other location vacated by some other item 'k' For making experimental comparisons with other heuristics, we have taken these stages in ascending order of the item numbers which retrievals are to be made

As earlier, if we make decision of retrieving an item at some stage, then the location from which the item will be retrieved may be different than the location when the decision would have been made to retrieve same item at some other stage Hence for the stages m+1, onwards the representation of the stages as locations will depend on the stage at which the decision for retrieval of that item has been made Hence from state to state at these stages the representation of that stage for storage location will vary Special attention is required here in computing the revenue function for a state at these stages

3.3 Computational Performance

For the exact as well as the heuristic methods discussed in the sections 3 2 1 and 3 2 2, some problem sets are designed and solved using computer All the experiments are done on HP9000 platform, using UNIX operating system Further in case a problem has not been solved within five minutes of computational time it is terminated

Following two types of problems are considered

A. Storage/Retrieval Without the Use of Just Vacated Location

B. Storage/Retrieval With the Use of Just Vacated Location

The parameters considered in the generation of problems are

a the total number of storage items and locations

b the total number of retrieval items and locations

For the purpose of generating problems, following factors as function of the above parameters are used

I The factor X is defined as

$$X = \frac{\textbf{no. of storing locations - no. of retrieving locations}}{\textbf{no. of storing locations}} * 100 \qquad \textbf{...factor I}$$

This is a normalised factor, which considers the difference of the number of retrieving locations from the number of the storing locations Three set of problems with different values of X are created The first one has values of X from 0 to 25, the second one has the values of X from 25 to 50 and third has the values of X more than 50

II The factor Y here can be defined as

$$Y = \frac{\textbf{no. of items to be stored}}{\textbf{no. of locations for storage}} + \frac{\textbf{no. of items to be retrieved}}{\textbf{no. of locations for retrieval}} \qquad \textbf{...factor II}$$

This is also a normalised factor, in which the average number of items to the number of locations available for decision making is considered The different values of Y considered for comparisons are 0 75, 0 80, 0 90, and 1 00

III Here the absolute factor Z has been considered for comparisons, which can be defined as

Z = no. of items to be stored + no. of items to be retrieved. ...factor III

This factor is considered for analysis of result by keeping in view that this factor truly reflects the number of available locations for storage as well as for retrieval and the number of stages for dynamic programming kind of situation The values of Z for which analysis is made are 10, 15, 20, 25, and 30

Following performance measures for comparison of heuristics have been used

A. Storage/Retrieval Without the Use of Just Vacated Location

In this situation we know the optimum solutions as well as the approximate solutions for different heuristics, and hence comparisons can be made for the deviation of the objective value function value for the heuristics from the optimal value This deviation of the results can be described as

$$\textbf{\% deviation from opt.} = \frac{\textbf{heuristic cost - opt. cost}}{\textbf{opt. cost}} * 100$$

Apart from this, the comparisons are also made for the relative deviation of the heuristics This deviation of solutions can be defined as

$$\textbf{\% deviation among heuristics} = \frac{\textbf{heur2 cost - heur1 cost}}{\textbf{min(heur1 cost, heur2 cost)}} * 100$$

The curves for different problem sets are plotted taking the % of experiments on vertical axis and deviations of results on horizontal axis Analysis based on different factors can be done as follows

factor I :- For this case, the assumption is made at the design stage of problems that an item exists at only one item, i e, there is only one bin for each of the item The experiments are conducted for eighty cases in which value of X has been kept in the ranges 0 - 25, 25 - 50 and more than 50 The conclusions from graphs can be drawn as follows

a) For % deviation from optimal

(i) As the value of X increases, the deviation of the objective function value from optimal for heuristic 1 goes on decreasing

(ii) The similar deviation in case of heuristic 2 has shown almost same trend in all the cases

(iii) In all the cases, heuristic 1 has lesser deviation than that of heuristic 2

b) For % deviation among heuristics

(i) As the value of X increases, the heuristics are showing converging trends, i e, the deviation among heuristics goes on decreasing with the increase in X

(ii) In all the cases, the heuristic 1 shows better performance than heuristic 2 Specially for lower value of X, heuristic 1 deviate more than heuristic 2, in better side

factor II :- We have designed hundred problems for this case and value of Y taken are 0.75, 0.80, 0 90, 1 00 The results obtained for different cases may be summarised as follows.

a) **For % deviation from optimal**

(i) For all values of Y, the deviation of heuristic 1 from optimal cost is lower than that of heuristic 2

(ii) Also with the increase in Y, heuristic 1 shows slightly increased deviation, whereas heuristic 2 shows almost similar characteristic in all the cases

b) For % deviation among heuristics

(i) For all the values of Y, heuristics are showing zero deviation for large percentage of experiments

(ii) Almost in all the cases, heuristic 1 is showing better performance than heuristic 2

factor III :- The number of experiments conducted is hundred and the value of Z used are 10, 15, 20, 25, and 30 The analytical summary for the results in different cases can be presented as follows

a) For % deviation from optimal

It can be observed from the graphs that for all Z, heuristic 1 as well as heuristic 2 is showing very small deviation from the optimum result, however at minute level, heuristic 1 still performs better than heuristic 2

b) For % deviation among heuristics

In this case also, the heuristics are showing very good convergence for lower value of Z, i e deviation of the objective function value for heuristic 1 from that of heuristic 2 is very low for a major part of the experiments However at higher value of Z, deviations get increased and that too in the better side of the heuristic 1

B. Storage/Retrieval With the Use of Just Vacated Location

Similar to previous case, in this case also we can compute the optimal cost as well as heuristic cost of the problem We have taken same problem set as in the previous set The

difference here with the previous case is that a retrieval location can be reused for the storage purpose

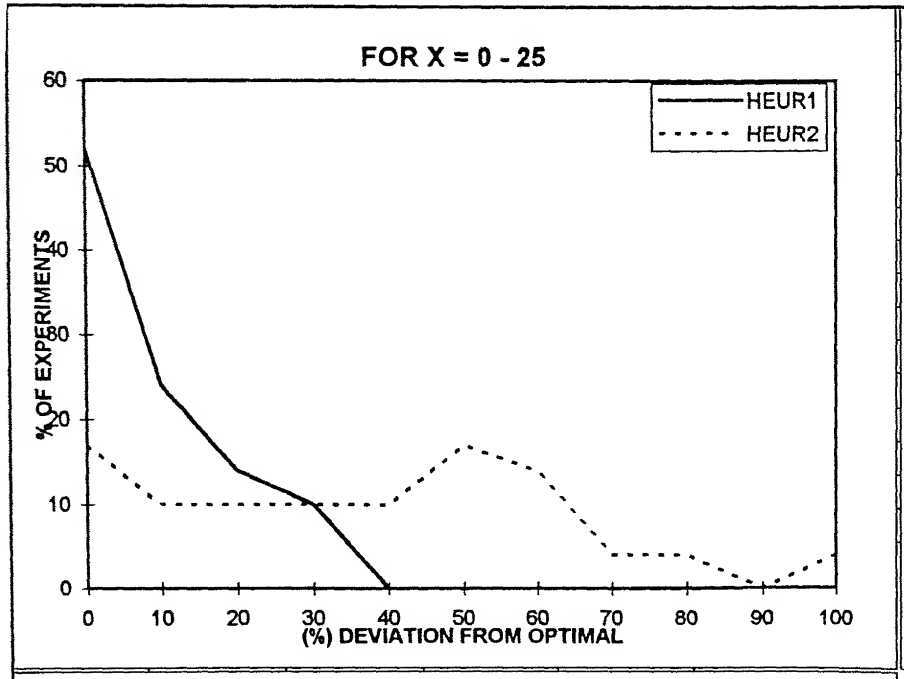
We have solved fifty problem using exact method as well as heuristic method It has been considered that the number of retrieval locations for an retrieval item is one The reason behind this simplification is that a problem with large number of storage locations and retrieval locations cannot be solved in computation time less than five minutes As shown in the figure 3 8, heuristic 1 (HVDP) is giving optimal result for large percentage of the problems, whereas heuristic 3 (HVEDP) is giving worst result among all heuristics Also the computational time taken by heuristic 3 is more than other heuristics for same problem Hence we can draw the conclusion that heuristic 3 is neither economical nor efficient for the given problem, and hence we close here the discussion on this heuristic

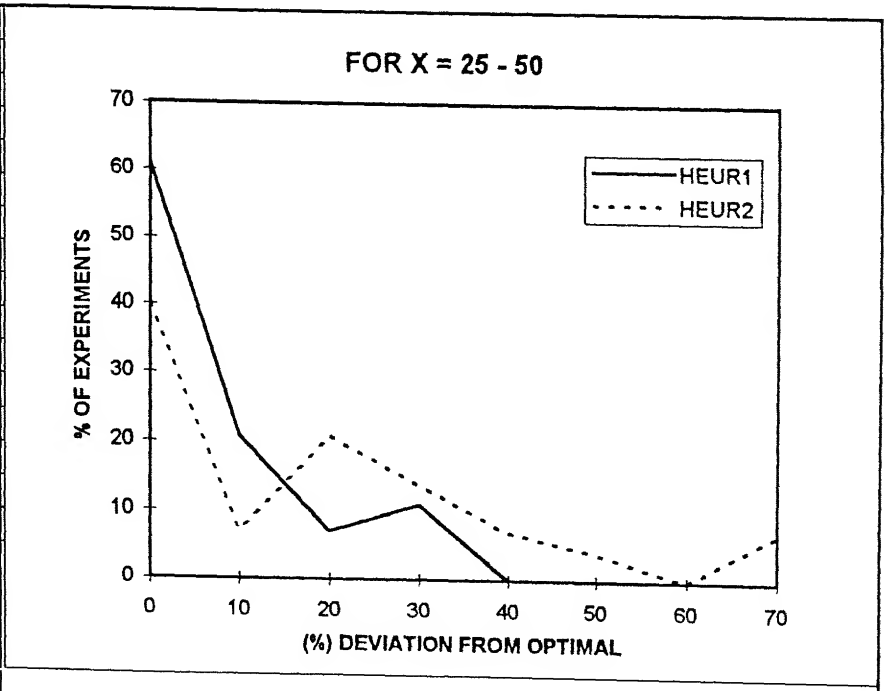
The same problem set as that in previous case has been used here also and also the comparisons are made for same factors As can be referred from the graphs for problem type A and type B, the nature of the graphs are same for the cases when the retrieval location is used for further storage and when it is not used for further storage, hence we can simply extend the conclusions for problem type A to this case Heuristic 1 (HVDP) is showing better performance in all the cases i e for all the factors considered here, than that of heuristic 2 (HVSP)

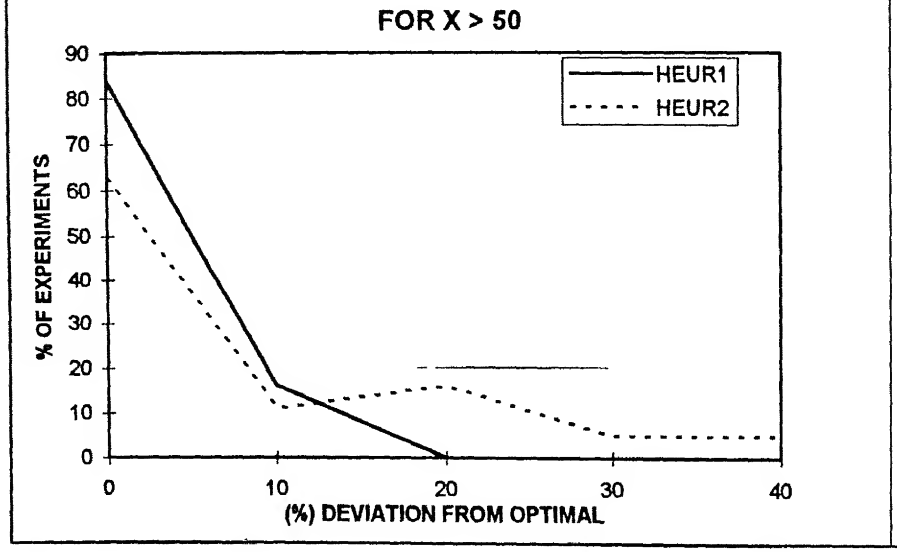
Apart from the comparisons of the objective value function, the comparisons are also made for the time, taken by different heuristics in solving the problem, when the factor III is used It can be concluded from the graph that as the time taken by heuristic 2 shows linear trend with the increase in the number of items to be stored and to be retrieved whereas heuristic 2 indicates exponential trend for the increase in same factor

Hence as the conclusion of the performance study, we can say that heuristic 1(HVDP) is very close to the optimal and is better than the heuristic 2(HVSP) However, the computational time required to solve the problem by heuristic 1 is much more than that by heuristic 2 for large problem, hence heuristic 2 can be recommended for large problems

comparison of heuristics with exact method for problem type A



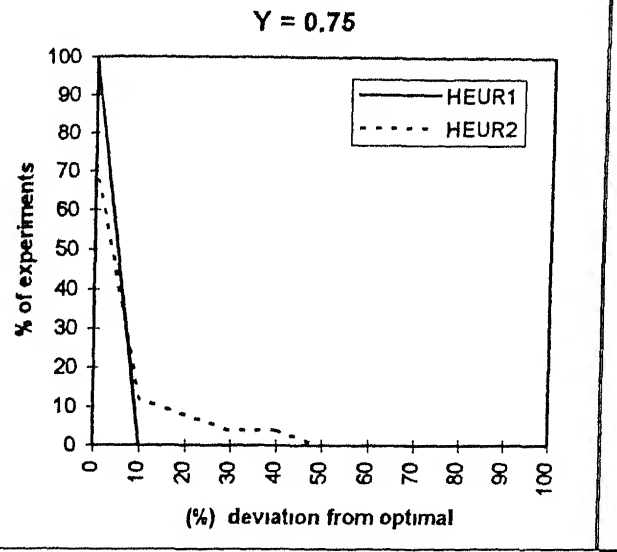


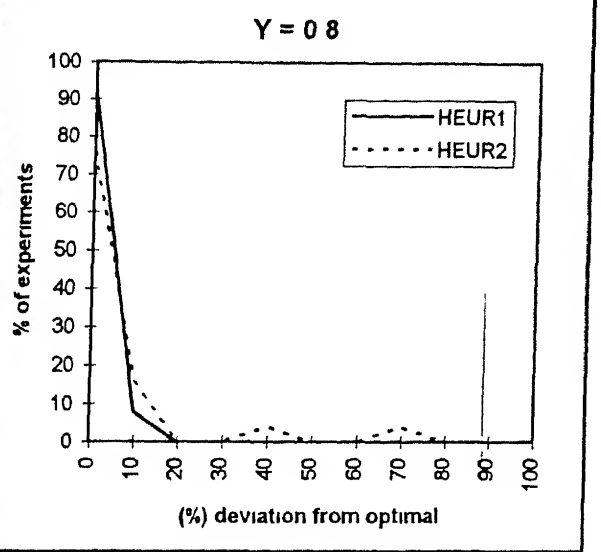


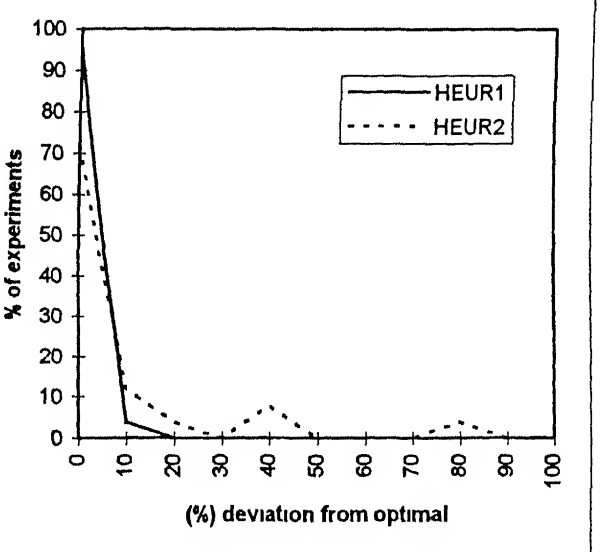
$$X = \frac{\text{storing locations} - \text{retrieving locations}}{\text{storing locations}} * 100$$

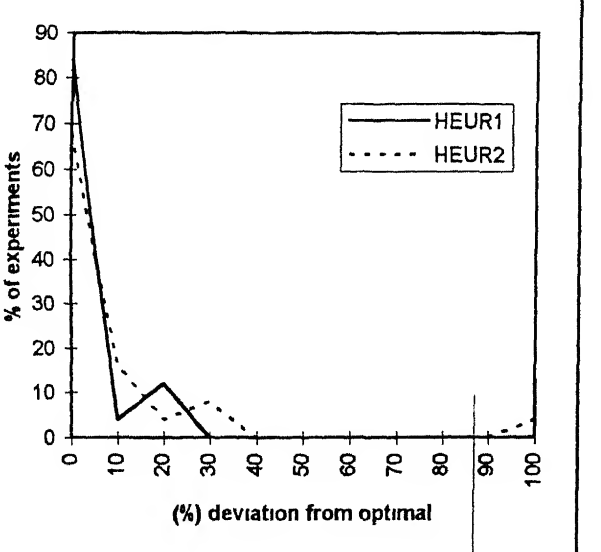
$$\% \text{ deviation} = \frac{\text{heur cost} - \text{opt cost}}{\text{opt cost}} * 100$$

comparison of heuristics with exact method for problem type A





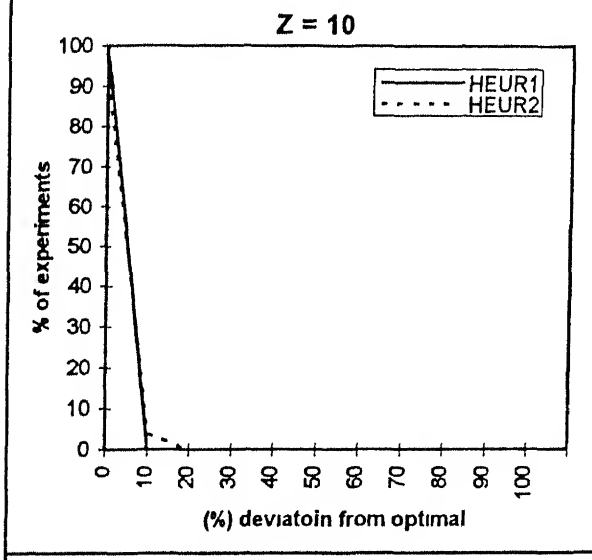


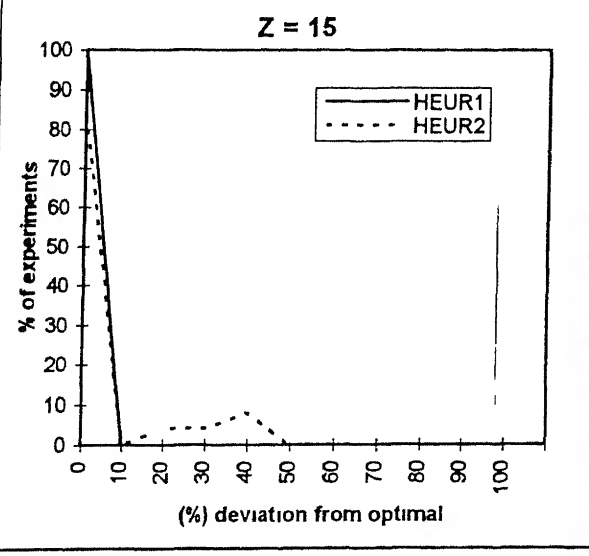


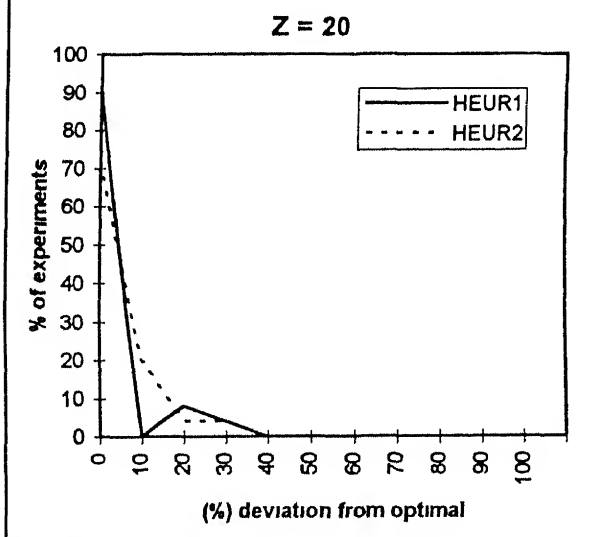
$$\%\ \text{deviation} = \frac{\text{heur cost} - \text{opt cost}}{\text{opt cost}} * 100$$

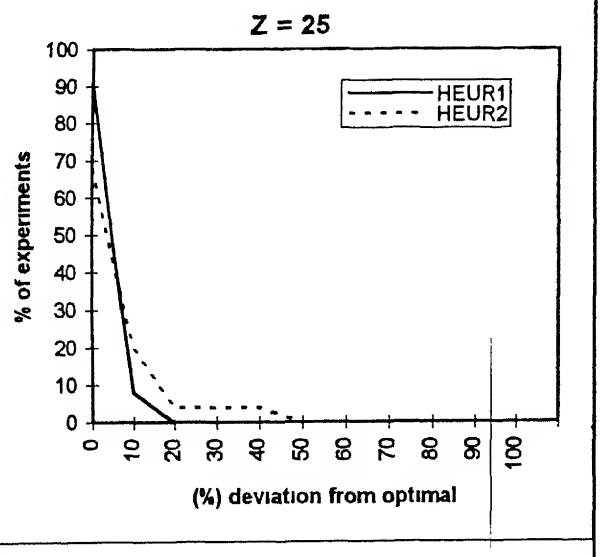
$$Y = \frac{\text{items to store}}{\text{locations to store}} + \frac{\text{items to retrieve}}{\text{locations to retrieve}}$$

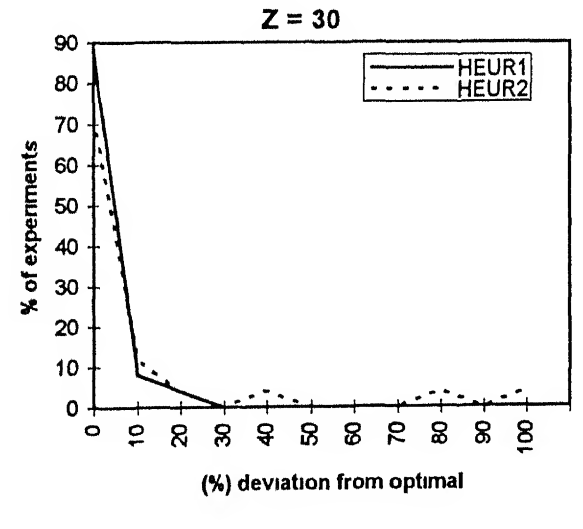
comparison of heuristics with exact method for problem type A







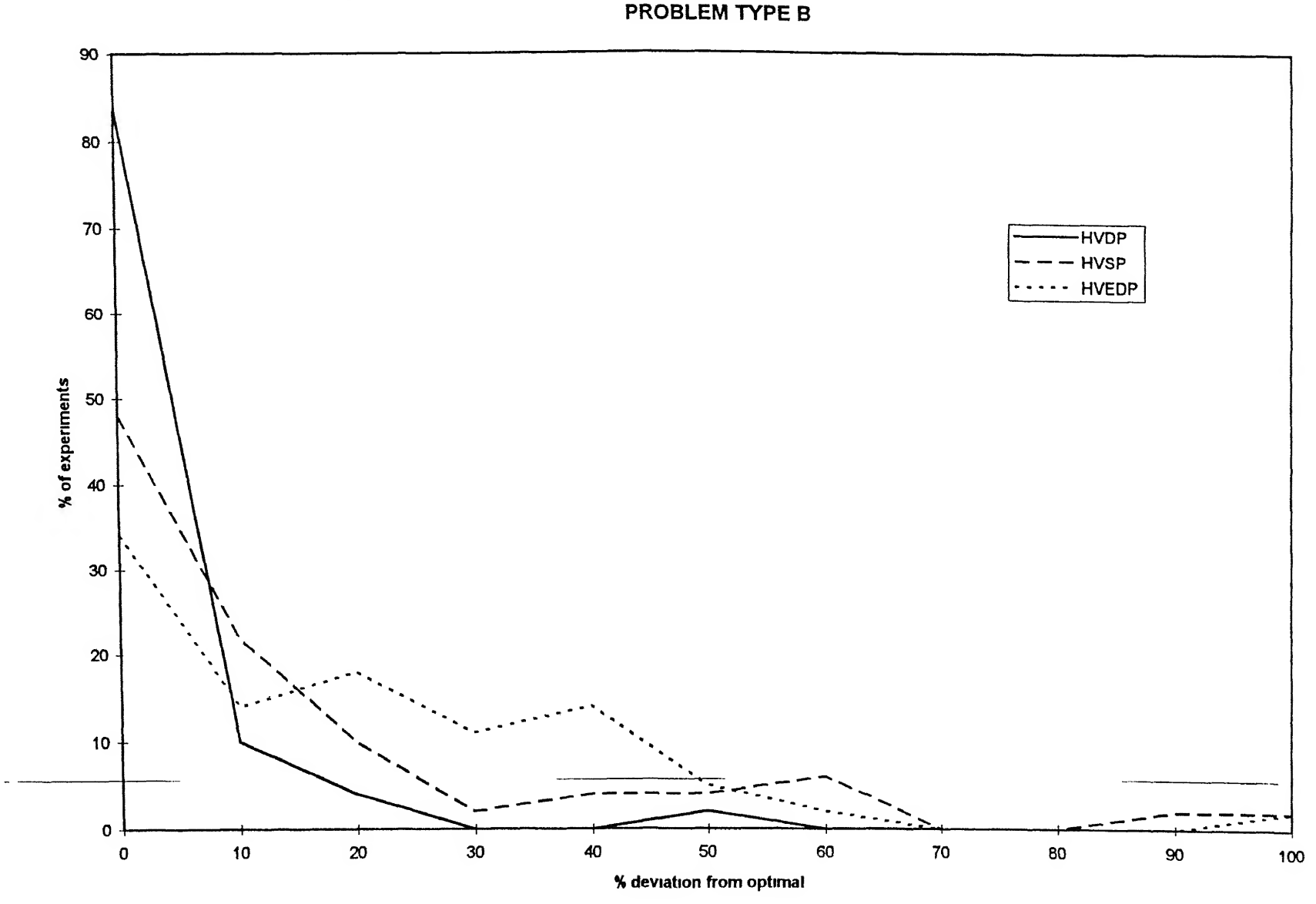




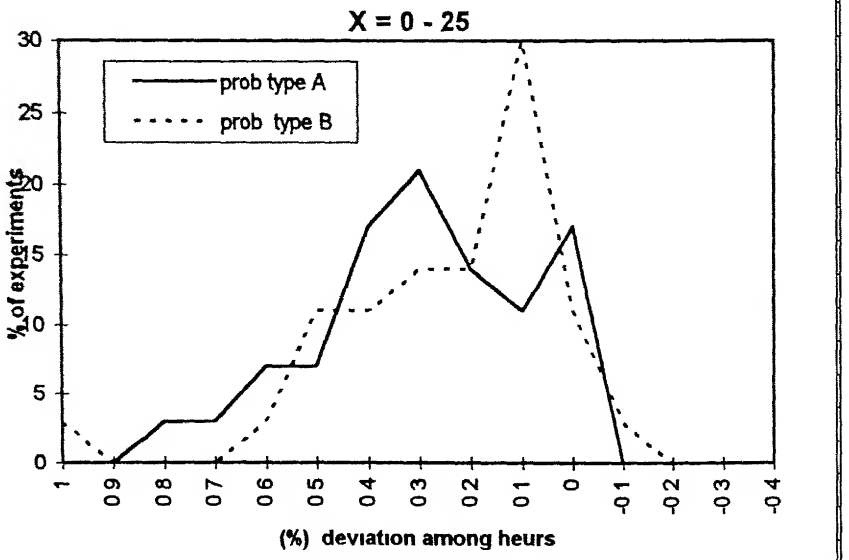
$$\% \text{ deviation} = \frac{\text{heur cost} - \text{opt cost}}{\text{opt cost}} * 100$$

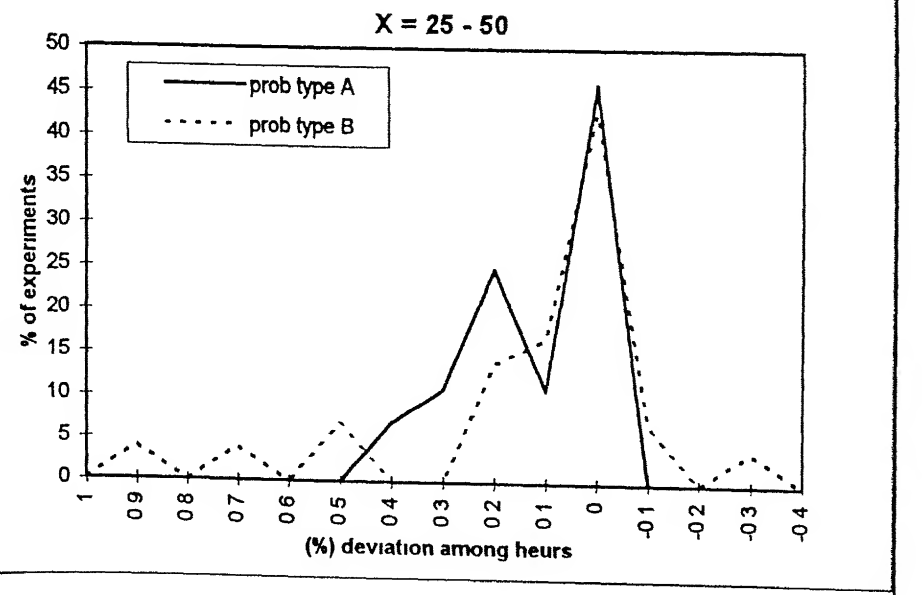
Z = items to store + items to retrieve

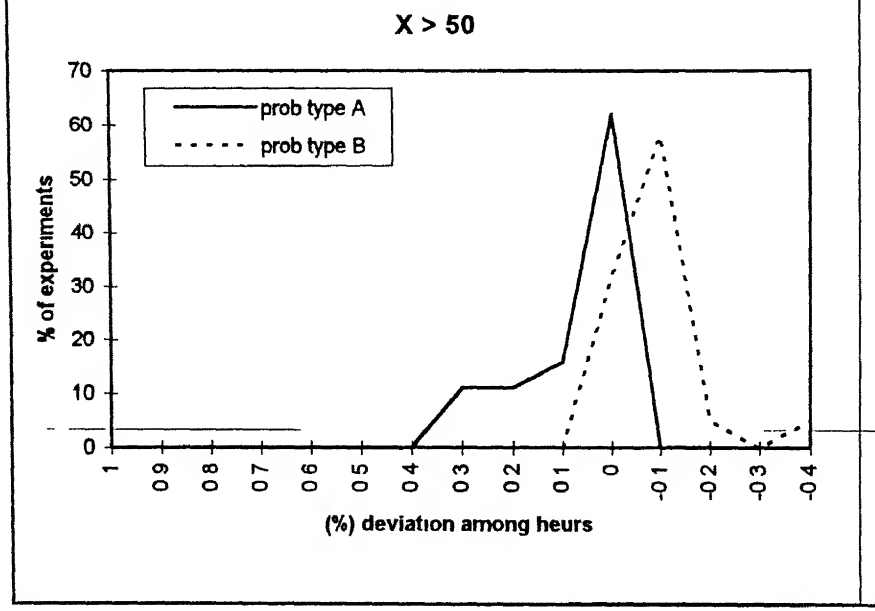
comparison of heuristics with exact method for problem type B



comparisons among heuristics for problem type A and B



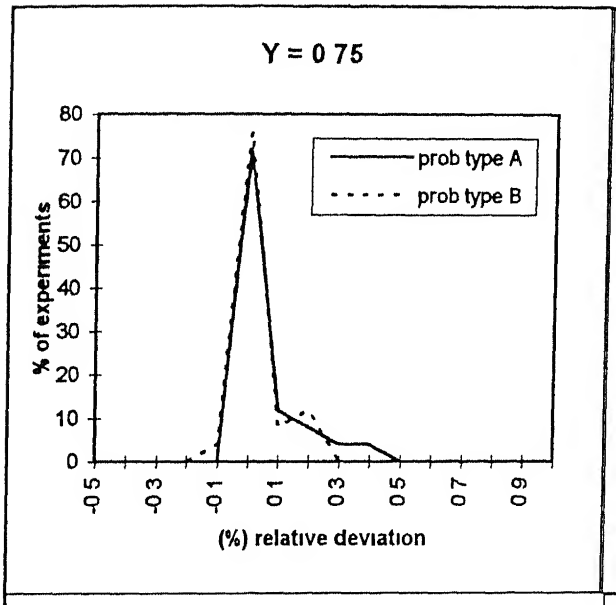


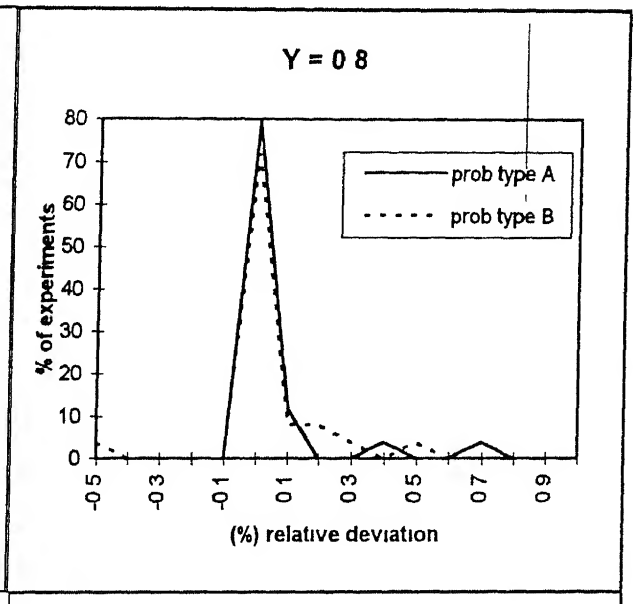


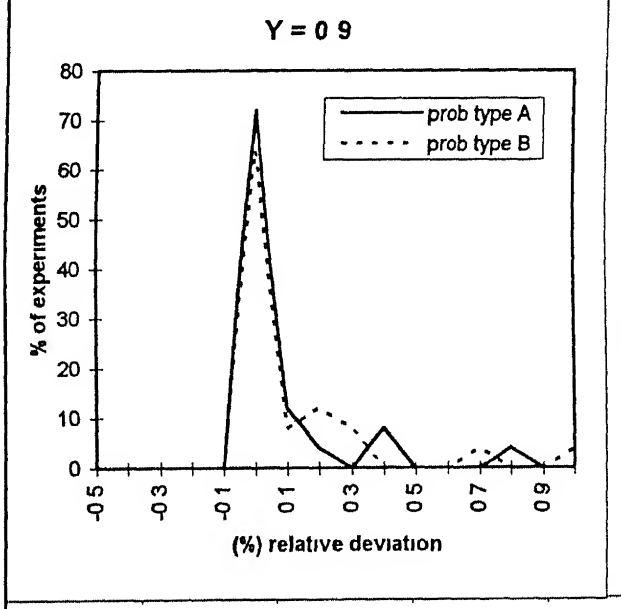
$$\%\ \text{deviation} = \frac{\text{heur2(cost)} - \text{heur1(cost)}}{\min(\text{heur1, heur2})} * 100$$

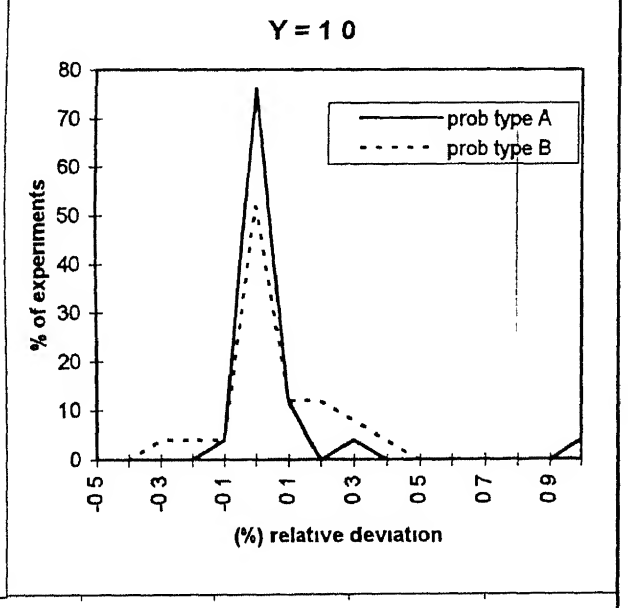
$$X = \frac{\text{storing locations} - \text{retrieving locations}}{\text{storing locations}} * 100$$

comparison among heuristics for problem type A and B





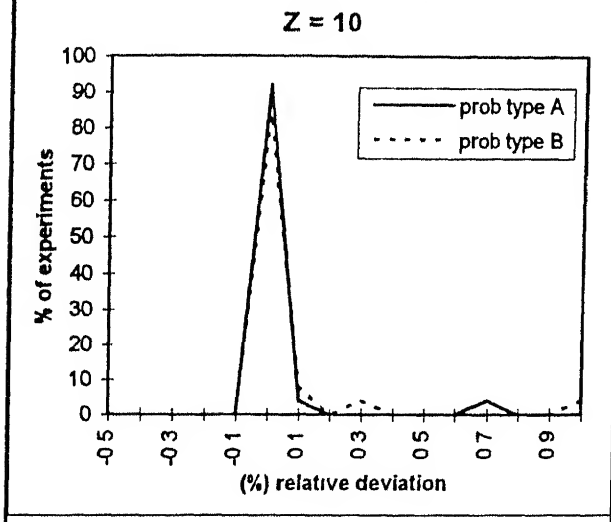


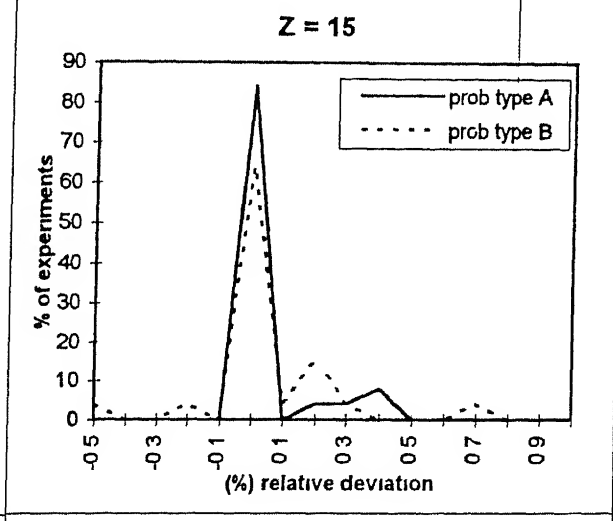


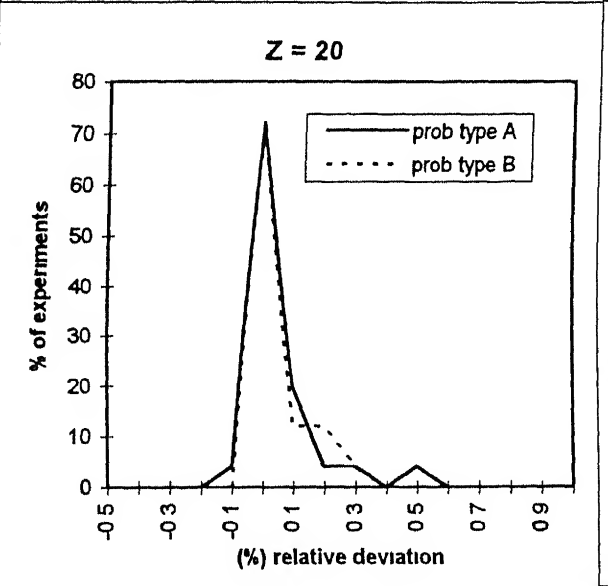
$$\% \text{ deviation} = \frac{\text{heur2 cost} - \text{heur1 cost}}{\min(\text{heur1, heur2})} * 100$$

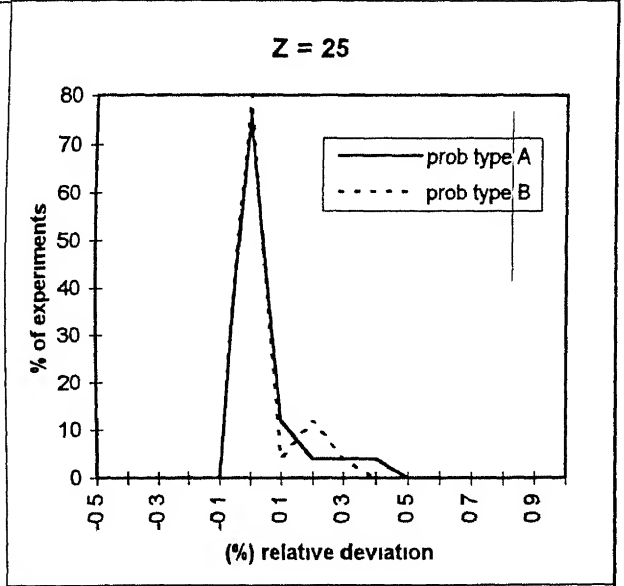
$$Y = \frac{\text{items to store}}{\text{locations to store}} + \frac{\text{items to retrieve}}{\text{locations to retrieve}}$$

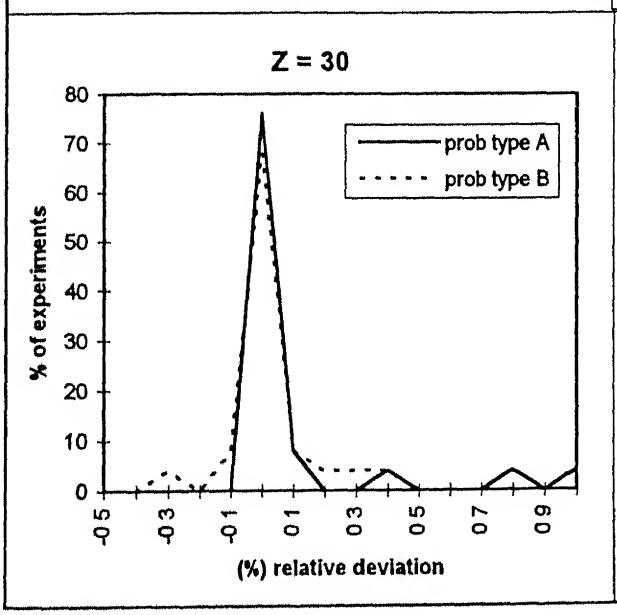
comparison among heuristics for problem type A and B







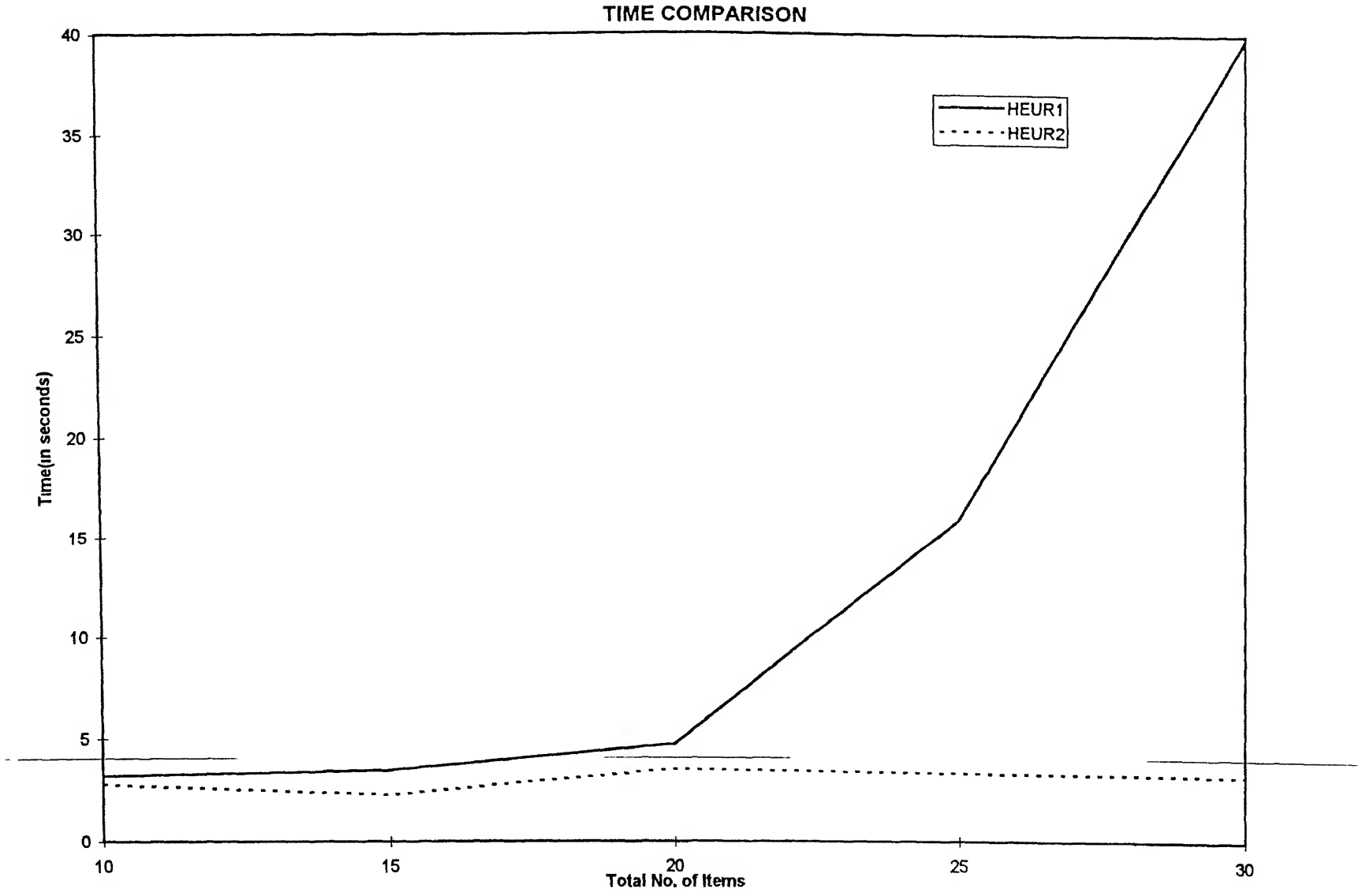




$$\% \text{ deviation} = \frac{\text{heur2 cost} - \text{heur1 cost}}{\min(\text{heur1, heur2})} * 100$$

Z = items to store + items to retrieve

Comparison of heuristics for problem type B



CHAPTER 4

MULTI-BIN STORAGE/RETRIEVAL SYSTEM

In this chapter, we are going to deal with the case, when the S/R machine is able to carry more than one bin at a time However the capacity of the S/R machine is fixed. Similar to the prior cases, the objective function which we have considered, is the minimization of the travel cost of the equipment Here we have considered that equipment will start its tour from input dock loaded with the number of bins which are to be stored, subject to the available capacity of the trolley Then it may go to store an item or to retrieve an item depending on the available capacity of the trolley It will continue to move from location to location, storing and retrieving items, till all the storage items are stored and all the items required to be retrieved are retrieved subject to capacity restriction, or trolley is full with the items to be retrieved and hence cannot retrieve any more items After completing required assignment it will go to the output dock and if needed will repeat the process

We have also assumed that the order in which items are to be stored is fixed apriori, as FCFS, but the retrieval order is not fixed at all and depends on the selection of retrieval location Besides these, it has been also assumed that the trolley is capable of doing storage and retrieval in any sequence, i e it is not necessary that first all storage should be done and then only retrieval can be done

The input to the problem is the number and identity of the items which are to be retrieved, the number of available storage locations from where retrieval can be done, the capacity of the trolley, the cost of travel from input dock to the locations, the cost of travel from locations to output dock and the inter - location travel costs associated with the problem Also the cost of travel from location 'i' to 'j', C_{ij}, is considered to be equal to

the cost of travel from location 'j' to 'i', C_{ji} Further a location specific cost C_i is assumed with each storage location

It has been considered that a retrieving item may exists at more than one locations, hence for retrieval of this item only one of these locations will be used Two types of problems are considered in this chapter One, when the location from where a item is retrieved in a tour can be used for the storage in the same tour of the equipment and, second when the location vacated so is not allowed to be used for the storage in the same tour of the equipment In this chapter we have discussed first problem first and then the second problem For each of the cases we have formulated an exact method and also have developed one heuristic

4.1. Storage/Retrieval When the Vacated Location is Used for Further Storage

In this case the equipment may not have to travel when it uses same location for retrieval as well as for storage purpose Hence the cost occurred in travelling may equal to zero, but there may be other costs associated in storing an item at a location, for example dissimilarity cost as mentioned in appendix A Thus in addition to travel cost we have also considered a cost associated with each of the locations, which is used for further storage in the same tour of the equipment The exact as well as heuristic for the problem are described in next sections

4.1.1. Exact Method(Integer Programming Formulation)

The problem here has been formulated as Integer Programme and is represented by fig 4 1 In the network shown below, the nodes 1 to n, represent the locations whereas node '0' and node 'n + 1' represent the input dock and output dock respectively The arcs from

node '0' to the other nodes has been kept directed, as the tour can't go to input dock, whereas the arcs between the nodes which represent the locations, are connected by undirected arc, which shows the movement of the equipment in between these locations in both directions Similar logic applies to the arcs from different nodes to the node 'n + 1' which represent output dock Further we restrict the formulation to the case when all storage and retrieval can be done in one tour If that is not possible, the process can be repeated after removing the storage/retrieval locations used in the tour

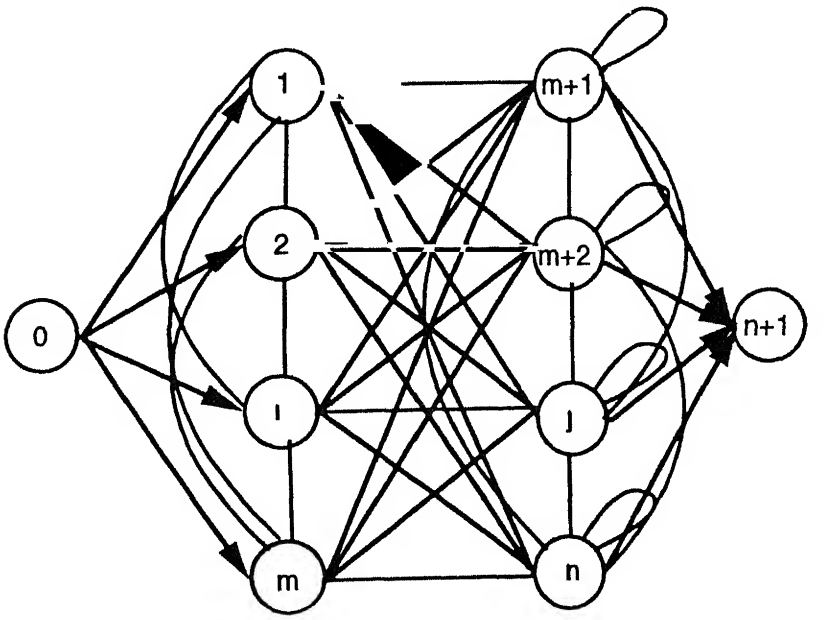


figure 4 1

Different notations used in the formulation of the problem can be represented as follows

Let,

i = 1, 2, , m are storing locations,

j = m + 1, m + 2, , n are retrieving locations,

c = capacity of the trolley,

a = number of items to be stored, (if c < a, a = c)

b = number of items to be retrieved, (if c < b, b = c)

$r_1, r_2, \ldots, r_b$ are the items to be retrieved,

$s_1, s_2, \ldots, s_a$ are the items to be stored,

S = {i, i is initially available storage location}

R_{r_i} = {k, 'k' is the location where item 'r_i' exists}

R = {l, l $\in R_{r_i}$ for all i = 1, 2, , b}
(it is the set of all retrieval locations)

I_i = {j, 'j' is the location which is connected by arc from location 'i',
j $\in (S \cup R)$} for i = 1, 2, , n

J_i = {i, if j $\in I_i$ for all i $\in (S \cup R)$} for j = 1, 2, , n

X_{ij}^t = 1, if equipment make tour from location 'i' to location 'j' at stage 't',

= 0, otherwise,

C_{ij} = cost of travel from location 'i' to location 'j',

y_i^t = 1, if the location 'i' is revisited for the storage at stage t, i = m+1, m+2, , n

C_i = fixed cost associated with location i when it is visited for the storage purpose

Hence I_i is the set of all those locations, to which the movement of trolley from location 'i' is allowed, and J_j is the set of all locations from which the trolley can move to 'j' in a tour These sets exclude the travel of the equipment from input dock to any location and similarly from any location to output dock Now by using above interpretation of the notations, the IP formulation of the problem can be represented as follows

Minimize,

$$\sum_{j=1}^{n} C_{0j} X_{oj}^1 + \sum_{t=2}^{a+b} \sum_{i=1}^{n} \sum_{j \in I_i} C_{ij} X_{ij}^t + \sum_{j=1}^{n} C_{j(m+n+1)} X_{j(m+n+1)}^{a+b+1} + \sum_{t=2}^{a+b} \sum_{i \in R} C_i Y_i^t + \sum_{i \in S} C_i (X_{0i}^1 + \sum_{t=2}^{a+b} \sum_{j \in J_i} X_{ji}^t) \quad (4\,1)$$

Subject to,

$$\sum_{i=1}^{n} \sum_{j \in I_i} X_{ij}^t = 1, \qquad t = 2, 3, \ , a+b, \quad (4\,2)$$

$$\sum_{t=2}^{a+b} \sum_{i \in R} Y_i^t + \sum_{i=1}^{m} X_{0i}^1 + \sum_{t=2}^{a+b} \sum_{i=1}^{m} \sum_{j=J_i} X_{ij}^t = a, \quad (4\,3)$$

$$\sum_{i=m+1}^{n} X_{0i}^{1} + \sum_{t=2}^{a+b} \sum_{i=m+1}^{n} \sum_{j \in J_i} X_{ji}^{t} - \sum_{t=2}^{a+b} \sum_{i=m+1}^{n} Y_i^t = b, \qquad (4\ 4)$$

$$\sum_{j=1}^{n} X_{0j}^{1} = 1, \qquad (4\ 5)$$

$$\sum_{j \in I_i} X_{ij}^{2} - X_{0i}^{1} = 0, \qquad i = 1,2, \quad , n \qquad (4\ 6)$$

$$\sum_{j \in I_i} X_{ij}^{t} - \sum_{j \in J_i} X_{ji}^{t-1} = 0, \qquad t = 3, 4, \quad ,a+b, \; i = 1,2, \quad , n, \qquad (4\ 7)$$

$$X_{i(n+1)}^{a+b+1} - \sum_{j \in J_i} X_{ji}^{a+b} = 0, \qquad i = 1,2, \quad , n, \qquad (4\ 8)$$

$$\sum_{j=1}^{n} X_{j(n+1)}^{a+b+1} = 1, \qquad (4\ 9)$$

$$c - a + \sum_{i=1}^{m} X_{0i}^{1} - \sum_{i=m+1}^{n} X_{0i}^{1} \geq 0, \qquad (4\ 10)$$

$$c - a + \sum_{i=1}^{m} X_{0i}^{1} + \sum_{k=2}^{t} \sum_{i=1}^{m} \sum_{j \in J_i} X_{ji}^{k} - \sum_{i=m+1}^{n} X_{0i}^{1} - \sum_{k=2}^{t} \sum_{i=m+1}^{n} \sum_{j \in J_i} X_{ji}^{k} + \sum_{k=2}^{t} \sum_{i=m+1}^{n} 2Y_i^t \geq 0,$$

$$t = 2,3, \quad ,a+b, \qquad (4\ 10)$$

$$\sum_{t=2}^{a+b} \sum_{j \in J_i} X_{ji}^{t} \leq 1, \qquad i = 1, 2, \quad , m, \qquad (4\ 11)$$

$$\sum_{t=2}^{a+b} \sum_{j \in J_i} X_{ji}^{t} \leq 2, \qquad i = m+1, m+2, \quad , n, \qquad (4\ 12)$$

$$X_{0i}^{1} + \sum_{k=2}^{t} \sum_{j \in J_i} X_{ji}^{k} - \sum_{k=2}^{t} Y_i^k \leq 1, \; i = m+1, m+2, \quad , n,$$

(4 13)

$$\sum_{i \in R_p} (X_{0i}^1 + \sum_{t=2}^{a+b} \sum_{j \in J_i} X_{ji}^t - \sum_{t=2}^{a+b} Y_i^t) = 1, p = 1, 2, \quad , b, \qquad (4\ 14)$$

$$X_{ij}^t = (0, 1), \qquad \text{for all arc } ij, \qquad (4\ 15)$$

$$Y_i^t = (0, 1), \qquad t = 2, 3, \quad , a+b, i = m+1, m+2, \quad , n \qquad (4\ 16)$$

The constraints in this formulation are explained below

(a) Constraint (4 2) ensures that the number of decisions taken at any step should be exactly equal to one

(b) Constraint (4 3) ensures that the required number of storage locations are selected

(c) Constraint (4 4) ensures that the number of retrieval equal to the required number,b

(d) Constraints (4 5), (4 6), (4 7), (4 8), and (4 9) ensure that there is an outgoing arc from a node if there is an incoming arc at that node in the previous stage, except for the starting node

(e) Constraints (4 10) and (4 11) take care of the capacity of the trolley, it implies that it will not carry more than its capacity at any stage i e number of unstored items on the trolley and items retrieved so far will not exceed the capacity of the same

(f) Constraint (4 12) implies that the equipment can visit to a location 'i', $i \in S$, only once

(g) Constraint (4 13) implies that the equipment can visit to a location 'j', $j \in R$, at most twice, once for retrieval of an item and second time for the storage of another item

(h) Constraint (4 14) implies that as soon as the equipment visit second time to a retrieval location, the variable Y_i corresponding to that location will take value 1

(i) Constraint (4 15) and (4 16) are integerality constraints

In the section 4 3, we have analysed the result of heuristic with this formulation, with the simplification in the formulation that an item 'j' exists at only one location This has been

necessary to keep the size of problem reasonable Hence in that case following changes will take place in the formulation

$n - m = b$,

$I_i = \{j, j = 1, 2, \quad , n \text{ if } i \in S \text{ then } i \neq j\}$,

$J_j = \{i, i = 1, 2, 3, \quad , n \text{ if } j \in S \text{ then } i \neq j\}$,

4.1.2. Approximate Method(Nearest - neighbour Search Technique)

The heuristic designed here for the problem is similar to the well known nearest neighbour heuristic for travelling salesman problem As discussed earlier, the tour of the equipment starts from the input dock and the method selects the least cost location from this point, for the equipment to travel The location thus selected may be storage location or retrieving location, depending upon the load and capacity of the equipment If the capacity of the equipment is more than the load on the equipment at a particular stage, then it can select a location for storage or a location for retrieval of the item, otherwise it will have to select only among the storage locations The process of selecting nearest location is done until and unless equipment completes its assignment and then travels to output dock This heuristic is simple to implement and also proved to be very efficient for a large category of problems Necessary modification has been made in the standard algorithm to incorporate the capability of storing at a location which has been emptied in the same tour of the equipment

The advantage of this method on earlier exact method is that if the capacity of the trolley is not able to assign all the items in a single tour, then we can reapply the same procedure for determining another tour of the equipment after deleting the used storage and retrieval locations from the list of storage and retrieval locations respectively, so that complete assignment can be done

4.2. Tour Without Revisiting a Retrieval Location

This case is a simplified version of the problem discussed earlier In this we don't use a location for further storage from where an item has been retrieved in the same tour of the equipment This can be stated as a special kind of TSP, when the tour is made through specified number of nodes Here we have considered same input to the problem as discussed earlier The problem has been formulated as Integer Programming for exact solution of the problem and similar methodology as previous case has been developed for heuristic solution of the problem These methods in short can be described as follows

4.2.1. Exact Method(Integer Programming Formulation)

The network representing the problem is shown by figure 4 2 As apparent, this network is similar to the previous one, except with the difference that we have not allowed the arcs 'ii' for any node in the network We have used same notations here as given in previous case

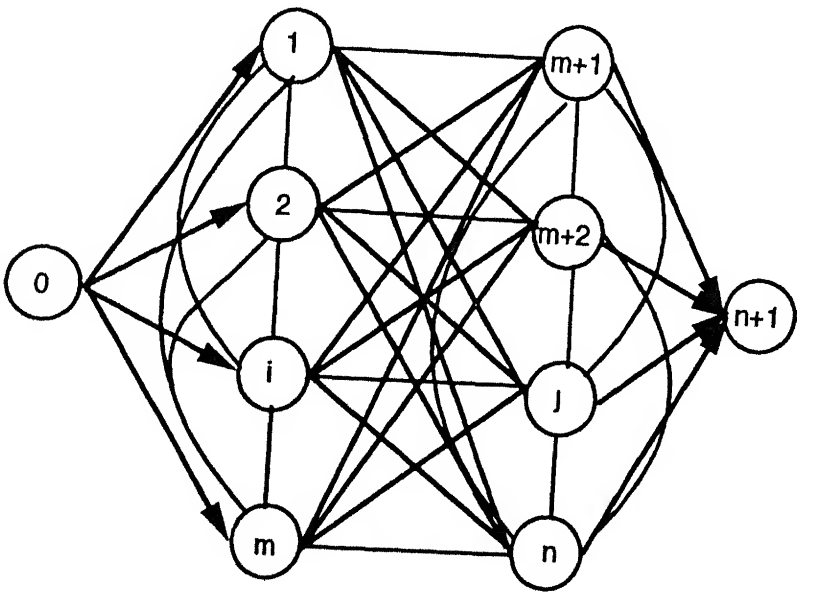


figure 4.2

With following changes in the earlier formulation the problem can be formulated as integer programme

(i) Make all $X_{ii}^{t} = 0$,

(ii) Make all $Y_{i}^{t} = 0$,

(iii) Change the constraint (4 12) from

$$\sum_{t=2}^{a+b}\sum_{j\in J_i} X_{ji}^{t} \leq 2, \qquad i = m+1, m+2, \ldots, n, \qquad (4\ 12)$$

to

$$\sum_{t=2}^{a+b}\sum_{j\in J_i} X_{ji}^{t} \leq 1, \qquad i = m+1, m+2, \ldots, n, \qquad (4\ 17)$$

This change in the constraint implies that a retrieved location can be visited only once at most and that also for retrieval purpose only

After making above changes the problem can be formulated as follows

Minimize,

$$\sum_{j=1}^{n} C_{0j} X_{0j}^{1} + \sum_{t=2}^{a+b}\sum_{i=1}^{n}\sum_{j\in I_i} C_{ij} X_{ij}^{t} + \sum_{j=1}^{n} C_{j(m+n+1)} X_{j(m+n+1)}^{a+b+1} + \sum_{i\in S} C_{i}\left(X_{0i}^{1} + \sum_{t=2}^{a+b}\sum_{j\in J_i} X_{ji}^{t}\right) \qquad (4\ 18)$$

Subject to,

$$\sum_{i=1}^{n}\sum_{j\in I_i} x_{ij}^{t} = 1, \qquad t = 2, 3, \ldots, a+b, \qquad (4\ 19)$$

$$\sum_{i=1}^{m} X_{0i}^{1} + \sum_{t=2}^{a+b}\sum_{i=1}^{m}\sum_{j=J_i} X_{ij}^{t} = a, \qquad (4\ 20)$$

$$\sum_{i=m+1}^{n} X_{0i}^{1} + \sum_{t=2}^{a+b}\sum_{i=m+1}^{n}\sum_{j\in J_i} X_{ji}^{t} = b, \qquad (4\ 21)$$

$$\sum_{j=1}^{n} X_{0j}^{1} = 1, \qquad (4\ 22)$$

$$\sum_{j\in I_i} X_{ij}^{2} - X_{0i}^{1} = 0, \qquad i = 1,2, \ldots, n \qquad (4\ 23)$$

$$\sum_{j\in I_i} X_{ij}^{t} - \sum_{j\in J_i} X_{ji}^{t-1} = 0, \qquad t = 3, 4, \quad ,a+b, \; i = 1,2, \quad , n, \tag{4 24}$$

$$X_{i(n+1)}^{a+b+1} - \sum_{j\in J_i} X_{ji}^{a+b} = 0, \qquad i = 1,2, \quad , n, \tag{4 25}$$

$$\sum_{j=1}^{n} X_{j(n+1)}^{a+b+1} = 1, \tag{4 26}$$

$$c - a + \sum_{i=1}^{m} X_{0i}^{1} - \sum_{i=m+1}^{n} X_{0i}^{1} \geq 0, \tag{4 27}$$

$$c - a + \sum_{i=1}^{m} X_{0i}^{1} + \sum_{k=2}^{t} \sum_{i=1}^{m} \sum_{j\in J_i} X_{ji}^{k} - \sum_{i=m+1}^{n} X_{0i}^{1} - \sum_{k=2}^{t} \sum_{i=m+1}^{n} \sum_{j\in J_i} X_{ji}^{k} \geq 0,$$
$$t = 2,3, \quad ,a+b, \tag{4 28}$$

$$\sum_{t=2}^{a+b} \sum_{j\in J_i} X_{ji}^{t} \leq 1, \qquad i = 1, 2, \quad , n, \tag{4 29}$$

$$\sum_{i\in R_p} (X_{0i}^{1} + \sum_{t=2}^{a+b} \sum_{j\in J_i} X_{ji}^{t}) = 1, \quad p = 1, 2, \quad , b, \tag{4 30}$$

$$0 \leq X_{ij}^{t} \leq 1, \qquad \text{for all arc } ij, \tag{4 31}$$

integers X_{ij}^{t} for all arc ij, (4 32)

Here different constraints have same interpretation as that in the earlier case, except we have no more the variables to take care the revisit of the equipment to a retrieving location The solution to this problem can be found by using 'CPLEX' We have made the

comparisons of heuristics with this method, but for the case when for each item to be retrieved there is only one retrieval location In that case

$I_i = J_j = \{k, k = 1, 2, \quad , n, i \neq j\}$

and $n - m = b$

4.2.2. Approximate Method(Nearest - neighbour Search Method)

This method is exactly similar to the heuristic described in the earlier case, except with the difference that there is no need of keeping track of the locations from where an item has been retrieved as no further storage can be made at these locations This heuristic is solved for a large set of problem The comparison with the exact method made only for smaller size problem with further restriction that each retrieval item has only retrieval location

4.3. Computational Performance

For the exact methods as well as heuristics discussed in the sections 4 1 and 4 2, some problem sets are designed and solved using computer All the computation works are done on HP9000 platform, using UNIX operating system Any problem taking computation time more than five minutes is terminated

Exact methods for all problem type are integer programme and are solved by using 'CPLEX' package available with AGNI

Following types of problems are used for the performance analysis

A. Storage/Retrieval Without the Use of Just Vacated Location

B. Storage/Retrieval With the Use of Vacated Location

The computational time required for solving the problems by exact methods increases exponentially with the increase in the size of the problem Hence we are unable to solve the problem larger that when altogether ten items to be stored/retrieved for problem type A, in the permitted time frame Similarly the problem size solved here for problem type B is kept below than twenty storage/retrieval for the same reason The reason behind such increase in computational time may be the multiple increase in the number of variables Hence all comparisons are done for the small size problems However heuristics are found efficient enough for solving the large size problems as well

The parameters used for the analysis of the results are the capacity of the trolley, number of items to be stored and the number of items to be retrieved For generating problems, following factor as function of the above parameters is used

$$X = \frac{\text{capacity of the trolley}}{\text{no of items to be stored + no of items to be retrieved}}$$

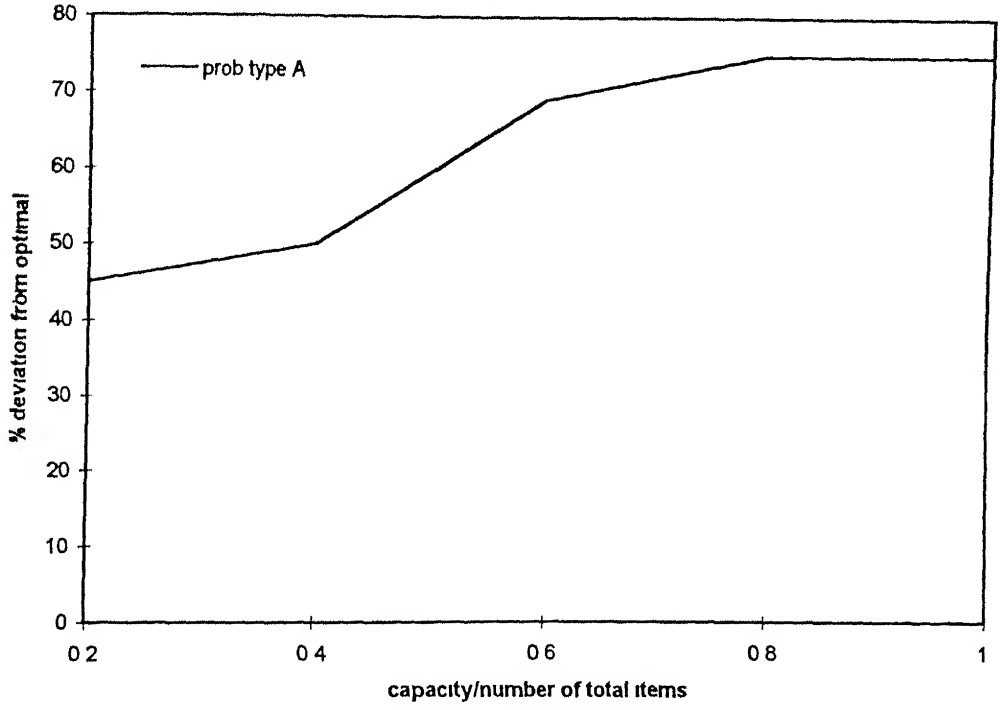
The values of factor for which analysis is made are 0 2, 0 4, 0 6, 0 8 and 1 0 The performance curves are shown in the figure 4 3 The ratio of the capacity of the trolley to the total number of items to be stored/retrieved is taken as horizontal axis and the percentage deviation of heuristic cost from optimal cost (defined below) is taken as vertical axis. The performance measurement of the heuristics with the exact method is done for following

$$\% \text{ deviation from optimal} = \frac{\text{heuristic cost - optimal cost}}{\text{optimal cost}}$$

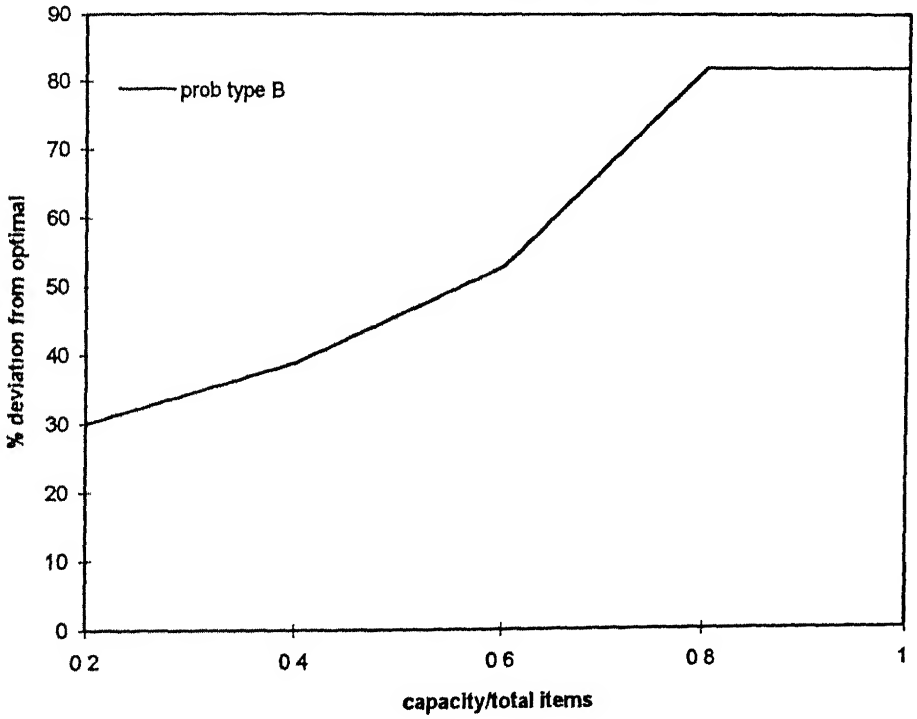
For each value of X ten different problems are solved by varying other parameters and the average of deviation of these results has been plotted as shown in figure 4 3

From the graph it can be observed that heuristic is showing relatively better performance in the case when the capacity of the equipment is low compared to the total number of items stored/retrieved As stated in section 4 1 2 the heuristic can also be used for determining all tours for complete assignment in such cases Hence it can be recommended that for the case when the capacity of the equipment is comparatively lower than the total number of items to be stored/retrieved, the heuristics can be used efficiently and effectively

comparison of heuristic with exact method for problem type A
(for single tour)



comparison of heristic with exact method for problem type B
(for single tour)



CHAPTER 5

CONCLUSIONS AND AVENUES FOR FUTURE WORK

5 1. Conclusions

In this dissertation storage and retrieval in a warehouse using dual command cycle is considered The problem is dealing with the dynamic storage retrieval system, i e the decisions about the storage locations and retrieval locations for the items are taken at the time the items come for the storage and retrieval In other words there is no fixed location for an item

Two type of material handling equipment has been considered leading to two types of problems In the first case the material handling equipment can carry only one bin, while in the second case it can carry more tahn one bin at the same time, limited by its capacity For both the cases two types of problem situation are considered In the first situation number of items to be stored and the number of items to be retrieved are one each and in the second situation there is a menu of storage items as well as of retrieval items Further for each of the problem situation two types of storage locations updating are considered In first type the location from which the retrieval is made can not be used for subsequent storage, while in the second type dynamic updating is made, which implies that location just vacated can be used for storage purpose

For each of the problem exact as well as heuristics are developed A large number of problems are solved on computer for each of the cases and the results are analysed The results analysis are done separately for different factors assuming that these factors may affect the performance of the methodologies In most of the cases it has been observed that although the exact solution of the problem is possible, the computational time required increases rapidly and hence heuristics have to be used in the practical situations

5.2. Avenues for Further Work

From the analysis of the different exact as well as heuristic methods, it has been observed that some of the heuristics are very good and can be applied to a practical situation, but there are the cases where further research can be done and better heuristics can be developed Further here assumption has been made that items are stored in prefixed order, as FCFS This assumption can be relaxed and the work can be done for the cases when the methodologies will have to decide storage as well retrieval order The on-line implementation of the methods developed here for practical cases can considered to be an avenue for future work

APPENDIX A

This section of the dissertation can be used to compute different cost functions associated with the storage and retrieval of the equipment In first part of this appendix we will deal with the computation of inter-aisle and intra-aisle distances In the second section we will deal with the dissimilarity cost associated with the storage of an item

A.1. Distance Function

The distance function between two locations 'i' and 'j' in a rack i e, d_{ij} is defined as the weighted horizontal, vertical, and across the aisle distance between them

$$d_{ij} = [H \{h|X_i - X_i|^a + e*\min[(X_i + X_j), (2*L - X_i - X_j)]\} + A|Y_i - Y_j|^b + V\{(f(Z_i + Z_j) + g|Z_i - Z_j|)^c\}]^d \quad (A\ 1)$$

Where

H is the weight for the horizontal distance

V is the weight for the vertical distance

A is the weight for the across the aisle distance

L is the length of rack along the aisle

X_i, X_j are the x-coordinates of locations 'i' and 'j' , along the aisle

Y_i, Y_j are the y-coordinates of locations 'i' and 'j', across the aisle

Z_i, Z_j are the z-coordinates of locations 'i' and 'j'(vertical movement)

a,b,c,d are the constants to determine the type of distance function, that is, rectilinear, Euclidean or squared Euclidean

f, g, and h are the dependent binary variables which take value 0 or 1 such that their sum does not exceed 1

Some of the possible cases of parameter settings are

(a) within the aisle movement

- equipment has to be brought down to the ground level before horizontal movement

  d = 1, e = 0, a = 1, f = 1, g = 0, c = 1

- equipment can move horizontally without changing the vertical height position

  d = 1, e = 0, a = 1, f = 0, g = 1, c = 1

- shortest distance movement between the location

  d = 1, a = 2, e = 0, g = 1, c = 2, d = 1/2

(b) across the aisle movement

h = 0, a = 1, c = 1, b = 1, f =1, g = 0, d = 1, e = 1

(c) shortest distance movement

h = 1, a = 2, e = 0, b = 2, f = 0, g = 1, c = 2, d = 1/2

So, if only rectilinear horizontal distance is to be included, while calculating the distance between two locations, keep the value of H, a, and d equal to one and weights V, A, b and c will take value zero Likewise, these weights take the values according to the requirement of the distance function.

A.2. Dissimilarity Cost

By dissimilarity cost here we mean that the cost associated with the storage of a particular item at a particular location This cost depends on the frequency with which the other items are retrieved with this item Hence by this a penalty can be given if two items, which are retrieved together frequently, are stored at two distant locations, i e it is an inverse function of the distance between two locations, from where items are retrieved together frequently

There are three possible cases in which the dissimilarity cost is computed These cases are discussed below along with the methodology of computing the dissimilarity cost

A.2.1 Single bin of an item is to be stored

In this case only one bin is to be stored at one location out of available m locations So this case is similar to single facility location in which one facility is to be located at one location out of some pre-specified number of available locations Here, a dissimilarity cost will be associated with each available location i Let the dissimilarity cost be denoted by a_i

$$a_i = \sum f_o * di(I_{i,o}) \quad \text{... A 2}$$

where

a_i = dissimilarity cost of location i

f_o = joint retrieval frequency of item o and item to be stored

d_{ij} = distance between locations i and j

$I_{i,o}$ = Location of item o closest to the location i

S = Set of all items in the warehouse

A.2.2 More than one bin of same item are to be stored

In this case, more than one bin of same item is to be located, so the interaction cost of locating the same item, at more than one location, will also be considered Let the total dissimilarity cost of locating the item at both the location i and j be denoted by a_{ij}

$$a_{ij} = a_i + a_j - p_{ij} \quad \text{A 3}$$

Where p_{ij} is the interaction term and is computed by the equation

$$p_{ij} = \Sigma f_o * b_{ij} * \text{Maximum}[d_{i,i(I_i,o)}, d_{j,(I_j,o)}] \quad \text{A 4}$$

Where $b_{ij} = 1$ if $I_{i,o} = I_{j,o}$

=0 otherwise

This interaction term is computed taking into consideration that in case, for any item "o", same location is closed to both the selected locations for the item being stored in it, it has to be associated with only one of them, which will be the location closest to it

A.2.3 More than one bin of different items are to be stored.

This case is similar to the case of multi-facility location in which new facilities to be located have some interaction among them So there will be a dissimilarity cost associated with each location for particular type of item and there will be an interaction factor for two type of items being located at two locations

Let

a_{ik} = dissimilarity cost of locating item i at location k

a_{jl} = dissimilarity cost of locating item j at location l

a_{ijkl} = dissimilarity cost of locating item i at location k and item j at location l

Then

$$a_{ijkl} = a_{ik} + a_{jl} - p_{ijkl} \qquad A\,5$$

where pijkl is the interaction factor mentioned above The expressions for the various costs mentioned above are given below The notations used here are same to that used in Equation A 2

$$a_{ik} = \Sigma\, f_{io} * d_{k(I_k,\, o)} \qquad A\,6$$

$$a_{jl} = \Sigma\, f_{jo} * d_{l\,(I_l,\, o)} \qquad A\,7$$

$$p_{ijkl} = \Sigma\, b_{kl} * \text{Maximum}[f_{io} * d_{k,\,(I_k,\, o)},\ f_{jo} * d_{l\,(I_l,\, o)}] \qquad A\,8$$

In this dissertation it has been assumed that items are stored on FCFS basis The order of decision making is assumed be the sequence in which items are to be stored Hence, the dissimilarity cost associated with an item for a particular location can easily be included in the travel cost of equipment to this location at a stage

BIBILIOGRAPHY

AGARWAL, K [1995], "Integereted Warehouse and Production System Stock Assignment and Order pick up", *Master's dissertation submitted to the Department of Industrial and Management Engineering, IIT kanpur*

ASHAYERI, J, and GELDERS, L F [1985]," Warehouse Design Optimization",*European Journal of Operational Research 21, 285-294*

AZAVIDAR, F [1987]," Optimum Allocation of resources Between the Random Access and Rack Storage Spaces in an Automated Warehousing System"

BANSAL, R [1995], "Integerated Warehouse and Production System - System Design and Implementation", *Master's dissertation submitted to the Department of Industrial and Management Engineering, IIT kanpur*

BOZER, Y A, and WHITE, J A [1984]," Travel-time Models for Automated Storage Retrieval Systems", *IIE Transactions 16(4), 329-338*

CHAIME, M E [1992]," Operations Sequencing in Automated Warehousing System", *International Journal of Production Research 30(10), 2401-2409*

CORMIER, G, and GUNN, E A [1992]," A Review of Warehouse Models", *European Journal of Operational Research 58, 3-13*

EGBELU, P J [1991]," Framework for Dynamic Positioning of Storage/Rtrieval Machines in an Automated Storage/Retieval System", *International Journal of Production Research 29(1), 17-37*

EGBELU, P J, and WU, C T [1993]," A Comparison of Dwell Point Rules in an Automated Storage/retrieval System", *International Journal of Production Research 31(11), 2515-2530*

ENYAN, A and ROSENBLATT, M J [1993]," An Interleaving Policy in Automated Storage/Retrieval System", *International Journal of Production Research 31(1), 1-18*

ENYAN, A and ROSENBLATT, M J [1994]," Establishing Zones in Single- Command Class - based rectangular AS/RS", *IIE Transactions, 38-45*

GOESTSCHALCKS, M , and RATLIFF, H D [1991],"Optimal Lane Depths for Single and Multiple Products in Block Stacking Storage System", *IIE Transactions 23/3, 245-258*

GOETSCHALCKS, M [1983]," Storage and Retrieval Policies for Efficient Order Picking Operations", *Ph D Thesis, Georgia Institute of Technology, Atlanta*

GOLDEN, B L , and ASSAD, A A [1988],"Vehicle Routing Methods and Studies", *North Holland, Amsterdam*

GRAVES, S C , HAUSMAN, W H , and SCHWARZ, L B [1977]," Storage Retrieving Interleaving in Automatic Warehousing System", *Management Sciences 23(9), 935-945*

HAN, M H , McGINNIS, L F , SHICK, J S , and WHITE, J A [1987]," On Sequencing Retrievals in an Automated Storage/Retrieval System", *IIE Transactions, 56-66*

HAUSMAN, W H , SCHAWRZ, L B and GRAVES, S C [1976]," Optimal Storage Assignment in Automatic Warehosing Systems", *Management Sciences 22(6), 629-638*

HWANG, H , and LIM, J M [1993]," Determining an Optimal Dwell Point of the Storage/Retrieval Machine in an Automated Storage/Retrieval Systems", *International Journal of Production Research 31(11), 2591-2602*

HWANG, H , and LEE, S B [1990]," Travel-time Models Considering the Operating Characteristics of the Storage and Retrieval Machine", *International Journal of Production Research 28(10), 1779-1789*

JAIKUMAR, R and SOLOMON, M M [1990]," Dynamic Operational Policies in an Automated Warehouse", *IIE Transactions 22(4), 370-376*

JAIN, R K [1995], "Simultaneous Location-Routing Problem for Stock Assignment in a Warehouse", *Master's dissertation submitted to the Department of Industrial and Management Engineering, IIT kanpur*

JARVIS, M , and McDOWELL, E D [1991]," Optimal Product Layout in an Order Picking Warehouse", *IIE Transactions 23(1), 92-102*

KIND, D A [1965]," Measuring Warehouse Space Utilisation", *Transportation and Distribution Management 15, 29-34*

LAPORTE, G , and NOBERT, Y [1981]," Effect of De-linking Locations Decisions with a Routing Decision", *European Journal of Operational Research 6, 224-226*

NAMBIAL, J M et al [1981]," A Large Scale Location-Allocation Problem in the Rubber Industry", *European Journal of Operational Research 6,183-189*

REDDY, N J M [1994], " Materail Handling Requirement Planning Using Parallel Machine Scheduling", *Master's dissertation submitted to the Department of Industrial and Management Engineering, IIT kanpur*

ROSENBLATT, M B [1994]," An Application of Cluster Analysis to the Problem of Locating Items within a Warehouse", *IIE Transactions 36, 101-103*

SRIVASTAVA, R , and BENTON, W C [1990],"The Location Routing Problem Considerations in Physical Distribution System Design", *Computers & Operations Research 17, 427-435*

TAHA, H A [1992]," Operations Research", *Marwell Mcmillan International, New York*

TOMPKINS, J A , and WHITE, J A [1984]," Facilities Planning", *(Newyork Wiley)*

TRETHEWAY, S J , and FOOTE, B L [1994]," Automatic Computation and Drawing of Facility Layouts with Logical Aisle Structures", *International Journal of Production Research 32, 1545-1555*